SURESH GYAN VIHAR UNIVERSITY

Bachelor of Computer Applications (BCA)

Program Project Report (PPR)

2024-25

Centre for Distance & Online Education (CDOE)

SURESH GYAN VIHAR UNIVERSITY
Accredited by NAAC with 'A+' Grade

Table of Contents

Program Mission and Objectives

Suresh Gyan Vihar University, Jaipur, established in 2008, is a leading private university of Rajasthan. SGVU, Jaipur is accredited with Grade A+ by National Assessment and Accreditation Council (NAAC), offers courses like Engineering, Management, hotel Management, Pharmacy, Arts, Humanities, Law, Agriculture, B.lib etc. in conventional mode. SGVU is renowned for its innovative academic practices, brilliance in technical education and consultancy to high profile industries. The program's mission is to impart, train and transform a student completely for high caliber competence through latest concepts and technology and equip the students as per the demands of the industry.

The program aims to achieve the following objectives

i. To provide an opportunity to get a BCA (Bachelor of Computer Applications) degree to those who find it difficult or even impossible to pursue regular BCA course at a university either due to their job commitments or certain other circumstances.

ii. To help the learners, study at their own pace, from their own chosen place.

iii. To provide students with an in-depth understanding of their chosen field of study, including current theories, research methodologies, and significant developments.

iv. To develop students' abilities to critically evaluate existing literature, arguments, and evidence within their field.

v. To encourage the integration of knowledge from various disciplines, promoting a more holistic understanding and innovative approaches to solving complex problems.

vi. To instill a strong sense of ethical responsibility and an understanding of the ethical implications of research and professional practice within their discipline.

Program Relevance with the University Mission & Goals

Suresh Gyan Vihar University (SGVU) was established with a vision to become a university with commitment to excellence in education, research and innovation aimed towards human advancement.

The proposed program is highly relevant to the SGVU's mission i.e.

- Facilitate holistic education through knowledge sharing, skilling, research, and

entrepreneurial development.

- Integrate academic and industrial collaborations towards nation's development.
- Mentor students' physical, mental, emotional, secular, and spiritual attributes to become a valued human resource as it aims to provide quality education to those aspiring candidates who are deprived of higher education due to the limited number of intakes in the conventional mode of education in the Universities.

Moreover, to keep the quality intact the curriculum and syllabus has been designed at par with the conventional mode keeping in mind the specific needs and acceptability of the learners' ODL mode and in keeping with the aims and objectives of the University also ensures the industry and future skills relevance.

Nature of Prospective Target Group of Learners

The curriculum of BCA is designed in such a way that it helps the students to become not only more employable but also encourages them to become entrepreneurs in the field of IT, ITES, etc. Primarily the target group of learners will be:

- Those deprived of admission in the regular mode due to limited intake capacity.
- Those employed in various organizations who desire to pursue higher education as a passion or as a means for movement up the promotional ladder.
- Drop outs primarily due to social, financial, and economic compulsions as well as demographic reasons.
- Population of any age and those living in remote areas where higher education institutes are not easily accessible.

Program Appropriateness for conduction in ODL mode

Conducting a Bachelor of Computer Applications (BCA) program in Open and Distance Learning (ODL) mode is highly appropriate and effective for acquiring specific skills and competencies in the field. The degree would be of most value to students which can support the development of critical thinking, research skills, and subject-specific knowledge. In various fields such as education, business, government sector and public administration, it provides professionals with the opportunity to acquire advanced theoretical knowledge and practical

skills that are directly applicable to their work environments.

PROGRAMME OUTCOMES (PO)

- PO 1: Apply knowledge of mathematics, computer science, and domain knowledge to solve real-world problems in various fields.
- PO 2: Identify, formulate, and analyze computing problems to define requirements and specifications appropriate to their solution.
- PO 3: Design, implement, and evaluate computer-based systems, processes, components, or programs to meet desired needs with realistic constraints.
- PO 4: Use current techniques, skills, and tools necessary for computing practices, and understand their limitations.

PROGRAMME SPECIFIC OUTCOMES (PSO)

- PSO 1: Develop a strong foundation in core areas of computer science and applications including programming, database management, and software engineering.
- PSO 2: Design, code, test, and debug software applications using appropriate programming languages and tools.
- PSO 3: Implement and manage databases, perform data analysis, and apply data mining techniques to extract meaningful insights.
- PSO 4: Design and develop responsive web and mobile applications using modern development frameworks and tools.
- PSO 5: Understand and apply principles of network design and security protocols to protect information systems.

Instructional Design

Curriculum Design

The curriculum is designed by experts in the field of computer science and have considered to include relevant topics that are contemporary and create environmental awareness. It is approved by the BoS (Board of Studies), the CIQA (Centre for Internal Quality Assurance), and the AC (Academic Council) of the university.

Faculty Requirement

Name of Program	Faculty Required	Faculty Available	Name of Faculty	Designation	Date of Joining
BCA	2	2	Dr. Aman Sharma	Assistant Professor	02/09/2023
			Dr. Sohit Agarwal	Assistant Professor	15/08/2024

Instructional Delivery:

- Interacting with learning materials (Hard Copy Textbooks)
- Delivery of Learning Materials through SLM
- Personal Mentor Available
- Personal Contact Programme (PCP) conducted at campus on Saturday and Sunday

Teaching and Examination Scheme for Bachelor of Computer Applications (2024-25)

(Applicable for both Regular Mode & Distance Mode Education)

Year – I | Semester – I **Semester: AUTUMN/PAVAS**

SN	Course Code	Course Name	Credits	Contact Hrs/Week			Exam Hrs.	Weightage (in %)	
				L	T/S	P		CIE	ESE
1	DCA101	Elementary Mathematics	4	4	0	0	3	30	70
2	DCA103	Introduction to IoT	4	4	0	0	3	30	70
3	DCA105	Principles of Programming using 'C'	4	4	0	0	3	30	70
4	DCA151	Principles of Programming using 'C' Lab	1	0	0	2	2	30	70
5	DCA107	Elementary Computers	4	4	0	0	3	30	70
6	DCA153	Office Automation Tools Lab	1	0	0	2	2	30	70
7	DCA109	English language - I	2	2	0	0	3	30	70
8	PC 101	Proficiency and Co-Curricular Activities – I	2					100	
		Total	20+2	18	0	4			

Note: *In ODL mode the counselling hours will be 12 hours for 4 credit courses. No credits of Proficiency and Co-Curricular Activities will be added for Distance Education.*

Teaching and Examination Scheme for Bachelor of Computer Applications (2024-25)

(Applicable for both Regular Mode & Distance Mode Education)

Year – I | Semester – II **Semester: SPRING/BASANT**

SN	Course Code	Course Name	Credits	Contact Hrs/Week			Exam Hrs.	Weightage (in %)	
				L	T/S	P		CIE	ESE
1	DCA110	English language - II	2	2	0	0	3	30	70
2	DCA102	Fundamentals of Operating System	4	4	0	0	3	30	70
3	DCA104	Programming in C++	4	4	0	0	3	30	70
4	DCA152	Programming in C++ Lab	1	0	0	2	2	30	70
5	DCA106	Python Programming	4	4	0	0	3	30	70
6	DCA154	Python Programming Lab	1	0	0	2	2	30	70
7	DCA108	Human Value & Ethics	1	2	0	0	3	30	70
8	DCA112	Environmental Studies	3	3	0	0	3	30	70
9	PC 102	Proficiency and Co-Curricular Activities – II	2					100	
		Total	20+2	19	0	4			

Note: *In ODL mode the counselling hours will be 12 hours for 4 credit courses. No credits of Proficiency and Co-Curricular Activities will be added for Distance Education.*

Teaching and Examination Scheme for Bachelor of Computer Applications (2024-25)

(Applicable for both Regular Mode & Distance Mode Education)

Year – II | Semester – III **Semester: AUTUMN/PAVAS**

SN	Course Code	Course Name	Credits	Contact Hrs/Week			Exam Hrs.	Weightage (in %)	
				L	T/S	P		CIE	ESE
1	DCA201	Financial Management	4	4	0	0	3	30	70
2	DCA203	Database Management System	4	4	0	0	3	30	70
3	DCA205	Web Architecture Design	4	4	0	0	3	30	70
4	DCA251	Web Architecture Design Lab	1	0	0	2	2	30	70
5	DCA207	Data Visualization Techniques	4	4	0	0	3	30	70
6	DCA253	Data Visualization Techniques Lab	1	0	0	2	2	30	70
7	DCA209	Programming in Java	4	4	0	0	3	30	70
8	DCA255	Programming in Java Lab	1	0	0	2	2	30	70
9	PC 201	Proficiency and Co-Curricular Activities – III	2					100	
		Total	23+2	20	0	6			

Note: *In ODL mode the counselling hours will be 12 hours for 4 credit courses. No credits of Proficiency and Co-Curricular Activities will be added for Distance Education.*

Teaching and Examination Scheme for Bachelor of Computer Applications (2024-25)

(Applicable for both Regular Mode & Distance Mode Education)

Year – II | Semester – IV **Semester: SPRING/BASANT**

SN	Course Code	Course Name	Credits	Contact Hrs/Week			Exam Hrs.	Weightage (in %)	
				L	T/S	P		CIE	ESE
1	DCA202	Computer Organization & Architecture	4	4	0	0	3	30	70
2	DCA204	Software Engineering Fundamentals	4	4	0	0	3	30	70
3	DCA206	Advanced Web Development	4	4	0	0	3	30	70
4	DCA252	Advanced Web Development Lab	1	0	0	2	2	30	70
5	DCA208	Data Structure & Algorithm	4	4	0	0	3	30	70
6	DCA254	Data Structure & Algorithm Lab	1	0	0	2	2	30	70
7	DCA210	Data Warehousing and Data Mining	4	4	0	0	3	30	70
8	PC 202	Proficiency and Co-Curricular Activities – IV	2					100	
		Total	22+2	20	0	4			

Note: *In ODL mode the counselling hours will be 12 hours for 4 credit courses. No credits of Proficiency and Co-Curricular Activities will be added for Distance Education.*

Teaching and Examination Scheme for Bachelor of Computer Applications (2024-25)

(Applicable for both Regular Mode & Distance Mode Education)

Year – III | Semester – V **Semester: AUTUMN/PAVAS**

SN	Course Code	Course Name	Credits	Contact Hrs/Week			Exam Hrs.	Weightage (in %)	
				L	T/S	P		CE	ESE
1	DCA351	Industrial Training / Internship	20	0	0	40	3	60	140
		Total	20	0	0	40			

Note: *In ODL mode the counselling hours will be 12 hours for 4 credit courses.*

Teaching and Examination Scheme for Bachelor of Computer Applications (2024-25)

(Applicable for both Regular Mode & Distance Mode Education)

Year – III | Semester – VI **Semester: SPRING/BASANT**

SN	Course Code	Course Name	Credits	Contact Hrs/Week			Exam Hrs.	Weightage (in %)	
				L	T/S	P		CIE	ESE
1	DCA302	Introduction to Sales Force	4	4	0	0	3	30	70
2	DCA304	Research and Intellectual Property & Rights	4	4	0	0	3	30	70
3	DCA306	Data Science Basic	4	4	0	0	3	30	70
4	DCA352	Data Science Basic Lab	1	0	0	2	2	30	70
5	DCA308	Software Testing	4	4	0	0	3	30	70
6	DCA354	Software Testing Lab	1	0	0	2	2	30	70
7	DCA356	Project	1	0	0	2	2	30	70
8	DCA358	Seminar	1	0	0	2	2	30	70
		Total	20	16	0	8			

Note: *In ODL mode the counselling hours will be 12 hours for 4 credit courses.*

L: lecture; T: Tutorial; S: Seminar; P: Practical

CIE: Continuous Internal Evaluation; ESE: End Semester Evaluation

SURESH GYAN VIHAR UNIVERSITY
Accredited by NAAC with 'A+' Grade

SYLLABUS

(SEMESTER-I)

Program: Bachelor of Computer Applications

Course Name: Elementary Mathematics	Course Code: DCA101
Semester: 1	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 4
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic knowledge of high school mathematics, including arithmetic, algebra, geometry, and basic trigonometry.

Course Objectives:

1. Understand the foundational concepts of mathematics for problem-solving in computer applications.
2. Learn set theory, logic, and matrix operations essential for computational logic.
3. Build the ability to solve linear and quadratic equations and their applications.
4. Develop analytical reasoning using probability and statistics.
5. Apply mathematical concepts to real-world problems in computing and IT.

Course Content:

Topic and Contents
BLOCK-1: Set Theory and Logic
Unit 1: Sets and Subsets Unit 2: Relations and Functions Unit 3: Propositional Logic
BLOCK-2: Algebra
Unit 4: Linear Equations and Inequalities Unit 5: Quadratic Equations Unit 6: Sequences and Series
BLOCK-3: Matrices and Determinants
Unit 7: Matrices Unit 8: Determinants Unit 9: Eigenvalues and Eigenvectors

BLOCK-4: Number Systems and Boolean Algebra
Unit 10: Number Systems Unit 11: Boolean Algebra Unit 12: Applications in Computing
BLOCK-5: Probability and Statistical
Unit 13: Probability Theory Unit 14: Descriptive Statistics Unit 15: Inferential Statistics

Course Outcomes:

1. Demonstrate basic arithmetic skills with whole numbers, fractions, and decimals.
2. Understand and apply foundational mathematical concepts and patterns.
3. Solve real-world problems using mathematical reasoning.
4. Interpret and represent data using charts, graphs, and tables.
5. Develop confidence and accuracy in performing mental and written calculations.

Reference:

1. Ministry of Education. (2023). Elementary mathematics syllabus for Grade 5. https://www.education.gov/math/grade5syllabus
2. National Council of Educational Research and Training (NCERT). (2022). Mathematics: Class 5 textbook. NCERT. https://ncert.nic.in/textbook
3. Department of Education. (2021). Grade 5 mathematics curriculum guide. Local Education Authority.
4. Smith, J. A., & Lee, R. T. (2020). Effective teaching strategies for elementary mathematics. Educational Publishers.
5. International Bureau of Education. (2019). Mathematics curriculum framework for primary education. UNESCO. https://www.ibe.unesco.org

Program: Bachelor of Computer Applications

Course Name: Introduction to IoT	Course Code: DCA103
Semester: 1	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 4
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic knowledge of computer networks, programming fundamentals, and an understanding of embedded systems.

Course Objectives:

1. Understand the basics of IoT and its applications across various domains.
2. Learn IoT architecture, protocols, and connectivity options.
3. Develop skills in designing and implementing IoT systems.
4. Explore IoT data management, security, and analytics.
5. Gain practical experience with IoT hardware, software, and cloud platforms.

Course Content:

Topic and Contents
BLOCK-1: Introduction to IoT
Unit 1: Basics of IoT and Applications Unit 2: IoT Ecosystem and Technologies Unit 3: IoT Reference Architecture
BLOCK-2: IoT Hardware and Software
Unit 4: IoT Hardware Platforms Unit 5: IoT Operating Systems and Middleware Unit 6: Programming for IoT

BLOCK-3: IoT Protocols and Connectivity
Unit 7: Communication Protocols (HTTP, MQTT, CoAP)
Unit 8: Wireless Technologies (Wi-Fi, Zigbee, Bluetooth)
Unit 9: IoT Network Design
BLOCK-4: IoT Data and Analytics
Unit 10: IoT Data Collection and Storage
Unit 11: Data Analytics in IoT
Unit 12: Cloud Platforms for IoT (AWS, Azure)
BLOCK-5: IoT Applications and Security
Unit 13: IoT for Smart Cities and Homes
Unit 14: Security Challenges in IoT
Unit 15: Case Studies in IoT

Course Outcomes:

1. Understand IoT principles, architecture, and applications.
2. Work with IoT hardware and software platforms to build IoT systems.
3. Implement communication protocols and design IoT networks.
4. Analyze IoT data and utilize cloud platforms for efficient processing.
5. Address security concerns and design secure IoT applications.

Reference:

1. Ashton, K. (2009). That 'Internet of Things' thing. RFID Journal. https://www.rfidjournal.com/that-internet-of-things-thing
2. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645–1660. https://doi.org/10.1016/j.future.2013.01.010
3. Bahga, A., & Madisetti, V. (2014). Internet of Things: A hands-on approach. VPT.
4. Vermesan, O., & Friess, P. (Eds.). (2014). Internet of Things: From research and innovation to market deployment. River Publishers.
5. Kortuem, G., Kawsar, F., Fitton, D., & Sundramoorthy, V. (2010). Smart objects as building blocks for the Internet of Things. IEEE Internet Computing, 14(1), 44–51. https://doi.org/10.1109/MIC.2009.143333333

Program: Bachelor of Computer Applications

Course Name: Principles of Programming using 'C'	Course Code: DCA105
Semester: 1	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 4
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic knowledge of computer systems, familiarity with problem-solving techniques, and understanding of simple mathematical concepts.

Course Objectives:

1. Learn the basic syntax and structure of the C programming language.
2. Understand the concepts of data types, variables, and operators in C.
3. Develop proficiency in using control structures for decision-making and loops.
4. Master functions, arrays, and pointers for efficient code writing.
5. Apply C programming skills to solve real-world problems through practical coding exercises.

Course Content:

Topic and Contents
BLOCK-1: Introduction to C Programming
Unit 1: Overview of C Programming Unit 2: Writing and Executing C Programs Unit 3: Structure of a C Program
BLOCK-2: Data Types and Operators
Unit 4: Primitive Data Types Unit 5: Operators in C Unit 6: Type Casting and Type Conversion
BLOCK-3: Control Structures
Unit 7: Conditional Statements (if, switch) Unit 8: Loops (for, while, do-while) Unit 9: Nested Loops and Conditional Statements

BLOCK-4: Functions and Arrays
Unit 10: Functions in C (Definition, Declaration, and Call) Unit 11: Arrays (1D and 2D) Unit 12: Passing Arrays to Functions
BLOCK-5: Pointers and Advanced Concepts
Unit 13: Introduction to Pointers Unit 14: Dynamic Memory Allocation Unit 15: Structures and File Handling

Course Outcomes:

1. Write, compile, and execute basic C programs.
2. Use variables, data types, and operators to perform calculations and data manipulation.
3. Implement control structures to create conditional logic and loops in programs.
4. Use functions and arrays to organize code and manage large datasets.
5. Understand and apply pointers and structures for dynamic memory management and file operations.

Reference:

1. Kernighan, B. W., & Ritchie, D. M. (1988). The C programming language (2nd ed.). Prentice Hall.
2. Balagurusamy, E. (2017). Programming in ANSI C (8th ed.). McGraw Hill Education.
3. Schildt, H. (2014). C: The complete reference (4th ed.). McGraw Hill Education.
4. Yashavant, P. K. (2021). Let us C (18th ed.). BPB Publications.
5. Deitel, H. M., & Deitel, P. J. (2016). C: How to program (8th ed.). Pearson Education.

Program: Bachelor of Computer Applications

Course Name: Principles of Programming using 'C' Lab	Course Code: DCA151
Semester: 1	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 1
Type of course: LAB	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of computers, familiarity with basic problem-solving, and some exposure to algorithms or logical reasoning.

Course Objectives:

1. Learn to write, debug, and execute basic C programs using a compiler.
2. Develop problem-solving skills by implementing algorithms in C.
3. Understand control structures and functions through practical coding exercises.
4. Master the use of arrays, pointers, and file handling through real-world coding problems.
5. Apply theoretical knowledge of C programming to solve practical problems.

Course Content:

Topic and Contents
Lab 1: Write a C program to display "Hello, World!" and explore basic input/output operations.
Lab 2: Implement arithmetic operations using operators and variables in C.
Lab 3: Write a program to demonstrate conditional statements (if, if-else, switch-case).
Lab 4: Implement loops in C using for, while, and do-while for iterative tasks.
Lab 5: Write a program to create and manipulate 1D and 2D arrays
Lab 6: Implement functions with arguments and return values in C.
Lab 7: Explore pointers and write a program for pointer arithmetic and array manipulation using pointers.
Lab 8: Demonstrate dynamic memory allocation using malloc(), calloc(), and free().
Lab 9: Write a program to define and use structures in C.
Lab 10: Implement file operations like reading, writing, and appending in C.

Course Outcomes:

1. Gain proficiency in writing and executing C programs.
2. Develop problem-solving abilities through algorithmic implementation.
3. Understand and apply control structures, functions, and arrays in programs.
4. Learn to use pointers, dynamic memory, and structures effectively.
5. Demonstrate file handling techniques to manage data through C programming.

Reference:

1. Kernighan, B. W., & Ritchie, D. M. (1988). The C programming language (2nd ed.). Prentice Hall.
2. Balagurusamy, E. (2017). Programming in ANSI C (8th ed.). McGraw Hill Education.
3. Schildt, H. (2014). C: The complete reference (4th ed.). McGraw Hill Education.
4. Yashavant, P. K. (2021). Let us C (18th ed.). BPB Publications.
5. Deitel, H. M., & Deitel, P. J. (2016). C: How to program (8th ed.). Pearson Education.

Program: Bachelor of Computer Applications

Course Name: Elementary Computers	Course Code: DCA107
Semester: 1	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 4
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of computers and an interest in learning about their fundamental concepts and applications

Course Objectives:

1. Understand the basic components and functions of a computer system.
2. Learn about different types of software and hardware used in computing.
3. Explore the concept of the internet and how computers are connected globally.
4. Gain knowledge about operating systems and file management.
5. Develop basic problem-solving skills through simple computer applications.

Course Content:

Topic and Contents
BLOCK-1: Introduction to Computers
Unit 1: Basic Introduction to Computer Unit 2: Components of a Computer System (Hardware & Software) Unit 3: Types of Computers (Desktops, Laptops, Servers, etc.)
BLOCK-2: Basic Computer Operations
Unit 4: Understanding Input and Output Devices Unit 5: Introduction to Operating Systems Unit 6: Using File Management Systems
BLOCK-3: Software Applications
Unit 7: Introduction to Word Processing Software Unit 8: Basics of Spreadsheet Software Unit 9: Introduction to Presentation Software

BLOCK-4: The Internet and Communication
Unit 10: Introduction to the Internet Unit 11: Using Web Browsers and Search Engines Unit 12: Email and Communication Tools
BLOCK-5: Basic Programming Concepts
Unit 13: Introduction to Programming Languages Unit 14: Writing Simple Programs (e.g., using Scratch or Python) Unit 15: Introduction to Algorithms and Problem Solving

Course Outcomes:

1. Understand the basic structure and functioning of computer systems.
2. Operate common computer software tools for everyday tasks.
3. Use the internet for research, communication, and information sharing.
4. Create simple computer programs and understand basic programming logic.
5. Apply knowledge of computer systems to solve simple problems in a real-world context.

Reference:

1. Shelly, G. B., & Vermaat, M. E. (2012). Discovering computers: Fundamentals (1st ed.). Cengage Learning.
2. Norton, P. (2006). Introduction to computers (6th ed.). McGraw Hill.
3. Goel, A. (2010). Computer fundamentals. Pearson Education.
4. Williams, B. K., & Sawyer, S. C. (2015). Using information technology: A practical introduction to computers & communications (11th ed.). McGraw Hill Education.
5. Rajaraman, V. (2018). Fundamentals of computers (6th ed.). PHI Learning.

Program: Bachelor of Computer Applications

Course Name: Office Automation Tools Lab	Course Code: DCA153
Semester: 1	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 1
Type of course: LAB	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of computers and operating systems.

Course Objectives:

1. Familiarize with office automation software tools.
2. Learn to use word processors, spreadsheets, and presentation software effectively.
3. Develop skills for creating and managing documents, spreadsheets, and presentations.
4. Learn to automate tasks using macros and basic scripting in office applications.
5. Apply the knowledge of office automation tools for productivity in real-world scenarios.

Course Content:

Topic and Contents
Lab 1: Introduction to Office Automation Tools (Microsoft Office, Google Workspace, etc.).
Lab 2: Creating and formatting a document in a word processor (e.g., Microsoft Word).
Lab 3: Performing text formatting, styles, and using templates in a word processor.
Lab 4: Creating tables, lists, and graphics in word processing software.
Lab 5: Using formulas and functions in spreadsheet software (e.g., Microsoft Excel).
Lab 6: Creating and formatting a chart in spreadsheet software (bar, pie, line, etc.).
Lab 7: Working with multiple worksheets and linking data in spreadsheet applications.
Lab 8: Develop a presentation with slides, images, and charts in presentation software (e.g., Microsoft PowerPoint).
Lab 9: Automating tasks with macros in Word or Excel.
Lab 10: Integrating office tools by embedding objects (charts, tables) from spreadsheets into Word or PowerPoint documents.

Course Outcomes:

1. Gain proficiency in using word processors, spreadsheets, and presentation tools.
2. Develop automated tasks using macros and simple scripts.
3. Create professional documents, reports, and presentations for various purposes.
4. Understand how to integrate and use different office tools for efficient work.
5. Apply office automation tools to enhance productivity and accuracy in office tasks.

Reference:

1. Shelly, G. B., & Vermaat, M. E. (2014). Microsoft Office 2013: Introductory. Cengage Learning.
2. Basandra, S. K. (2014). Computer applications in management. I.K. International Publishing House.
3. Goyal, A. (2017). MS Office 2016 training guide. BPB Publications.
4. Friedlein, L. (2016). Mastering office automation tools: A comprehensive guide. McGraw Hill Education.
5. Lambert, J., & Cox, J. (2015). Microsoft Word 2016 step by step. Microsoft Press.

Program: Bachelor of Computer Applications

Course Name: English language - I	Course Code: DCA109
Semester: 1	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 1
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic knowledge of English grammar and vocabulary.

Course Objectives:

1. Develop listening, speaking, reading, and writing (LSRW) skills in English.
2. Enhance proficiency in effective verbal and non-verbal communication.
3. Build confidence in public speaking and professional communication.
4. Improve academic and workplace writing skills.
5. Develop an understanding of intercultural communication.

Course Content:

Topic and Contents
BLOCK-1: Fundamentals of Communication
Unit 1: Basics of Communication Unit 2: Verbal and Non-verbal Communication Unit 3: Barriers to Communication
BLOCK-2: Listening and Speaking Skills
Unit 4: Active Listening Techniques Unit 5: Conversations and Dialogues Unit 6: Public Speaking and Presentations
BLOCK-3: Reading Skills
Unit 7: Skimming and Scanning Unit 8: Reading Comprehension Unit 9: Interpreting Visual Data

BLOCK-4: Writing Skills
Unit 10: Basic Grammar and Sentence Formation
Unit 11: Writing Formal Emails and Letters
Unit 12: Report and Summary Writing
BLOCK-5: Professional and Intercultural Communication
Unit 13: Workplace Communication Skills
Unit 14: Group Discussions and Interviews
Unit 15: Intercultural Communication

Course Outcomes:

1. Communicate effectively in English in academic, social, and professional settings.
2. Demonstrate active listening and effective public speaking skills.
3. Read and comprehend a variety of texts with ease.
4. Write clear and concise academic and workplace documents.
5. Exhibit confidence in intercultural and professional communication.

Reference:

1. Crystal, D. (2003). The Cambridge encyclopedia of the English language (2nd ed.). Cambridge University Press.
2. Yule, G. (2020). The study of language (7th ed.). Cambridge University Press.
3. Quirk, R., Greenbaum, S., Leech, G., & Svartvik, J. (1985). A comprehensive grammar of the English language. Longman.
4. Murphy, R. (2019). English grammar in use: A self-study reference and practice book for intermediate learners of English (5th ed.). Cambridge University Press.
5. Swan, M. (2016). Practical English usage (4th ed.). Oxford University Press.

SYLLABUS

(SEMESTER-II)

Program: Bachelor of Computer Applications

Course Name: English language - II	Course Code: DCA110
Semester: 2	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 1
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic proficiency in English language skills, including reading, writing, listening, and speaking.

Course Objectives:

1. Enhance communication skills in professional and academic settings.
2. Improve understanding of language structure and usage.
3. Develop reading, writing, listening, and speaking skills for effective expression.
4. Strengthen the ability to analyze, interpret, and summarize information.
5. Cultivate skills for professional communication, including writing reports and delivering presentations.

Course Content:

Topic and Contents
BLOCK-1: Language Fundamentals
Unit 1: Parts of Speech Unit 2: Sentence Structure and Types Unit 3: Tenses and Verb Usage
BLOCK-2: Effective Writing Skills
Unit 4: Paragraph Writing and Structure Unit 5: Writing for Different Purposes (e.g., essays, letters, articles) Unit 6: Editing and Proofreading
BLOCK-3: Critical Thinking and Reading
Unit 7: Identifying Key Ideas and Supporting Details Unit 8: Analyzing Arguments and Evidence Unit 9: Developing Critical Reading Skills

BLOCK-4: Communication in Professional Contexts
Unit 10: Writing Professional Emails and Memos Unit 11: Report Writing and Proposal Creation Unit 12: Delivering Presentations and Public Speaking
BLOCK-5: Advanced Language Skills
Unit 13: Idiomatic Expressions and Phrasal Verbs Unit 14: Developing Fluency in Spoken English Unit 15: Interpreting and Analyzing Visual and Audio Content

Course Outcomes:

1. Demonstrate proficiency in both written and spoken English.
2. Develop the ability to write clear, concise, and effective texts for various contexts.
3. Enhance critical reading and thinking skills for better comprehension and analysis.
4. Communicate effectively in professional and academic environments.
5. Apply advanced language skills, including understanding idioms and expressions, to improve fluency.

Reference:

1. Carter, R., & McCarthy, M. (2006). *Cambridge grammar of English: A comprehensive guide to spoken and written English grammar and usage*. Cambridge University Press.
2. Halliday, M. A. K., & Matthiessen, C. M. I. M. (2014). *An introduction to functional grammar* (4th ed.). Routledge.
3. Thornbury, S. (2005). *Beyond the sentence: Introducing discourse analysis*. Macmillan Education.
4. Biber, D., Johansson, S., Leech, G., Conrad, S., & Finegan, E. (1999). *Longman grammar of spoken and written English*. Longman.
5. Trudgill, P., & Hannah, J. (2017). *International English: A guide to the varieties of Standard English* (6th ed.). Routledge.

Program: Bachelor of Computer Applications

Course Name: Fundamentals of Operating System	Course Code: DCA102
Semester: 2	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 4
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of computer hardware and software concepts.

Course Objectives:

1. Understand the fundamental concepts and functions of an operating system.
2. Learn about process management, memory management, and file systems.
3. Explore the role of the operating system in multitasking and resource allocation.
4. Gain insights into system security, user interfaces, and system calls.
5. Understand the practical applications of operating systems in modern computing.

Course Content:

Topic and Contents
BLOCK-1: Introduction to Operating Systems
Unit 1: Basic Introduction to Operating System Unit 2: Functions and Types of Operating Systems Unit 3: Evolution of Operating Systems
BLOCK-2: Process and Memory Management
Unit 4: Process Concept and Scheduling Unit 5: Inter-process Communication and Synchronization Unit 6: Memory Management Techniques (paging, segmentation)
BLOCK-3: File Systems and I/O Management
Unit 7: File System Concepts and Structure Unit 8: File Allocation Methods Unit 9: I/O Management and Device Drivers

BLOCK-4: Operating System Security and Protection
Unit 10: Security Issues in Operating Systems Unit 11: Access Control Mechanisms Unit 12: Authentication and Authorization
BLOCK-5: Advanced Operating System Concepts
Unit 13: Distributed Operating Systems Unit 14: Virtual Machines and Cloud Computing Unit 15: Case Study: UNIX and Windows Operating Systems

Course Outcomes:

1. Understand the basic functions and components of an operating system.
2. Apply process scheduling and memory management techniques.
3. Implement file management and I/O techniques in an operating system.
4. Understand the importance of security and protection in OS design.
5. Explore the structure and functioning of advanced operating systems like UNIX and Windows.

Reference:

1. Silberschatz, A., Galvin, P. B., & Gagne, G. (2020). *Operating system concepts* (10th ed.). Wiley.
2. Tanenbaum, A. S., & Bos, H. (2014). *Modern operating systems* (4th ed.). Pearson Education.
3. Stallings, W. (2018). *Operating systems: Internals and design principles* (9th ed.). Pearson.
4. Deitel, H. M., Deitel, P. J., & Choffnes, D. R. (2004). *Operating systems* (3rd ed.). Pearson Education.
5. Bovet, D. P., & Cesati, M. (2005). *Understanding the Linux kernel* (3rd ed.). O'Reilly Media.

Program: Bachelor of Computer Applications

Course Name: Programming in C++	Course Code: DCA104
Semester: 2	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 4
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of programming concepts, such as variables, operators, and basic control structures, preferably in C or another language.

Course Objectives:

1. Understand the fundamentals of object-oriented programming (OOP) using C++.
2. Learn syntax and structure specific to C++ programming.
3. Implement classes, objects, and OOP principles such as inheritance and polymorphism.
4. Master advanced concepts like templates, file handling, and exception handling.
5. Develop problem-solving skills to implement real-world applications using C++.

Course Content:

Topic and Contents
BLOCK-1: Introduction to C++
Unit 1: Basics of C++ Programming Unit 2: C++ Syntax and Data Types Unit 3: Input/ Output in C++
BLOCK-2: Control Structures and Functions
Unit 4: Decision-Making and Loops Unit 5: Functions in C++ (Inline, Overloading) Unit 6: Recursion and Storage Classes
BLOCK-3: Object-Oriented Programming Concepts
Unit 7: Classes and Objects Unit 8: Constructors and Destructors Unit 9: Operator Overloading

BLOCK-4: Advanced OOP Concepts
Unit 10: Inheritance and Types of Inheritance Unit 11: Polymorphism and Virtual Functions Unit 12: Templates and Generic Programming
BLOCK-5: File Handling and Exception Handling
Unit 13: File Operations in C++ (Text and Binary Files) Unit 14: Exception Handling in C++ Unit 15: Standard Template Library (STL)

Course Outcomes:

1. Demonstrate proficiency in writing structured and object-oriented programs in C++.
2. Use control structures, functions, and classes to solve computational problems.
3. Apply OOP principles such as encapsulation, inheritance, and polymorphism.
4. Handle advanced programming features like templates, exceptions, and file I/O.
5. Develop efficient and reusable code using the Standard Template Library (STL).

Reference:

1. Stroustrup, B. (2013). *The C++ programming language* (4th ed.). Addison-Wesley.
2. Lippman, S. B., Lajoie, J., & Moo, B. E. (2012). *C++ primer* (5th ed.). Addison-Wesley.
3. Balagurusamy, E. (2017). *Object-oriented programming with C++* (8th ed.). McGraw Hill Education.
4. Deitel, H. M., & Deitel, P. J. (2017). *C++ how to program* (10th ed.). Pearson Education.
5. Meyers, S. (2014). *Effective modern C++: 42 specific ways to improve your use*

Program: Bachelor of Computer Applications

Course Name: Programming in C++ Lab	Course Code: DCA152
Semester: 2	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 1
Type of course: LAB	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of computer systems and programming fundamentals.

Course Objectives:

1. Develop hands-on experience with C++ programming concepts and syntax.
2. Apply object-oriented programming principles through practical exercises.
3. Master advanced features like operator overloading, templates, and file handling.
4. Enhance debugging and problem-solving skills by working on C++ programs.
5. Design and implement efficient solutions for real-world problems using C++.

Course Content:

Topic and Contents
Lab 1: Write a program demonstrating input/output operations and arithmetic calculations.
Lab 2: Implement a program using conditional statements and loops to solve a mathematical problem (e.g., factorial, Fibonacci sequence).
Lab 3: Write a program demonstrating inline functions, function overloading, and recursion.
Lab 4: Design a program to create and use classes, objects, and member functions.
Lab 5: Implement a class with constructors (default, parameterized, and copy) and a destructor to manage resources.
Lab 6: Create a program to overload operators (e.g., +, -, *) for custom data types.
Lab 7: Implement a program using virtual functions to achieve runtime polymorphism.
Lab 8: Demonstrate dynamic memory allocation using malloc(), calloc(), and free().
Lab 9: Write a program using templates for generic programming and implement exception handling for runtime errors.
Lab 10: Implement file operations like reading, writing, and appending in C.++

Course Outcomes:

1. Write, debug, and execute C++ programs efficiently.
2. Understand and implement object-oriented principles through practical coding tasks.
3. Work with advanced programming constructs like file handling, templates, and polymorphism.
4. Develop robust and reusable code by applying OOP and generic programming techniques.
5. Solve real-world problems using C++ with an emphasis on efficiency and modularity.

Reference:

1. Stroustrup, B. (2013). *The C++ programming language* (4th ed.). Addison-Wesley.
2. Lippman, S. B., Lajoie, J., & Moo, B. E. (2012). *C++ primer* (5th ed.). Addison-Wesley.
3. Balagurusamy, E. (2017). *Object-oriented programming with C++* (8th ed.). McGraw Hill Education.
4. Deitel, H. M., & Deitel, P. J. (2017). *C++ how to program* (10th ed.). Pearson Education.
5. Meyers, S. (2014). *Effective modern C++: 42 specific ways to improve your use of C++11 and C++14*. O'Reilly Media.

Program: Bachelor of Computer Applications

Course Name: Python Programming	Course Code: DCA106
Semester: 2	Core / Elective: Core
Teaching Scheme in Hrs (L: T:P): 3:0:0	Credits: 4
Type of course: Lecture + Assignments	Total Contact Hours:12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of programming concepts and logic. Familiarity with basic computer operations.

Course Objectives:

1. To introduce students to the fundamental concepts and syntax of Python programming.
2. To teach the use of Python data types, flow control statements, and functions.
3. To provide hands-on experience with pattern matching, regular expressions, and file handling.
4. To explore web scraping technique sand data extraction from websites.
5. To enable students to work with external data, such as Excel spreadsheets, and automate data handling tasks.

Course Content:

Topic and Contents
BLOCK-1: Introduction to Python Basics
Unit 1: Introduction To Basics Python Unit 2: Data Types Unit 3: Flow Control Operators / Statements
BLOCK-2: Functions in Python
Unit 4: Introduction To Functions In Python Unit 5: The Global Statement Unit 6: The List Data Type
BLOCK-3: Pattern Matching in Python

Unit 7: Pattern Matching With Regular Expressions Unit 8: Reading And Writing Files Unit 9: Organizing Files
BLOCK-4: Web Scraping
Unit 10: Web Scraping (Project: Mapit.Py With The Web Browser Module) Unit 11: Working With Excel Spreadsheet Unit 12: Updating A Spreadsheet
BLOCK-5: Advanced Python Concepts
Unit 13: Introduction to Object-Oriented Programming in Python Unit 14: Exception Handling in Python Unit 15: Working with Python Libraries

Course Outcomes

1. Students will be able to write Python programs using basic syntax and data types.
2. Students will be proficient in using flow control statements and function sin Python.
3. Students will demonstrates kill sin pattern matching and regular expressions for text processing.
4. Students will be able to perform file I/O operations, organize files, and manage directories.
5. Students will gain practical experience in web scraping and automating data manipulation with Python, including working with Excel spreadsheets

Reference:

1. Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. CreateSpace Independent Publishing Platform.
2. Lutz, M. (2013). *Learning Python* (5th ed.). O'Reilly Media.
3. Beazley, D. M. (2009). *Python cookbook* (3rd ed.). O'Reilly Media.
4. Al Sweigart. (2015). *Automate the boring stuff with Python: Practical programming for total beginners*. No Starch Press.
5. Downey, A. B. (2015). *Think Python: How to think like a computer scientist* (2nd ed.). O'Reilly Media.

Program: Bachelor of Computer Applications

Course Name: Python Programming LAB	Course Code: DCA154
Semester:2	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 1
Type of course: LAB	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic knowledge of programming concepts.

Course Objectives:

1. Understand and implement Python syntax, data types, and control structures.
2. Develop problem-solving skills using Python.
3. Implement functions and modules for code organization.
4. Explore file handling and error handling in Python.
5. Gain experience in working with Python libraries and frameworks.

Course Content:

Topic and Contents
Lab 1: Write a program to display "Hello World" and demonstrate the use of variables and data types.
Lab 2: Develop a program to check if a number is prime or not using conditional statements.
Lab 3: Create a Python program to generate Fibonacci numbers up to a specified range using loops
Lab 4: Write a function to compute the factorial of a number using recursion.
Lab 5: Implement a Python program to read a file and display its content.
Lab 6: Write a Python program that accepts a list of numbers and calculates the sum and average.
Lab 7: Create a Python program to demonstrate exception handling (e.g., handling division by zero).
Lab 8: Develop a program that accepts a sentence and counts the number of vowels and consonants in it.

Lab 9: Implement a Python program that uses dictionaries to store student names and their corresponding grades.
Lab 10: Build a Python application using a module to solve a real-world problem (e.g., a simple to-do list app).

Course Outcomes:

1. Write Python programs using basic syntax and data structures.
2. Implement Python control structures like loops and conditionals.
3. Design and use functions to structure Python programs.
4. Handle errors and exceptions in Python programs.
5. Work with files, collections, and libraries in Python.

Reference:

1. Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. CreateSpace Independent Publishing Platform.
2. Lutz, M. (2013). *Learning Python* (5th ed.). O'Reilly Media.
3. Beazley, D. M. (2009). *Python cookbook* (3rd ed.). O'Reilly Media.
4. Al Sweigart. (2015). *Automate the boring stuff with Python: Practical programming for total beginners*. No Starch Press.
5. Downey, A. B. (2015). *Think Python: How to think like a computer scientist* (2nd ed.). O'Reilly Media.

Program: Bachelor of Computer Applications

Course Name: Human Values & Ethics	Course Code: DCA108
Semester: 2	Core / Elective: Core
Teaching Scheme in Hrs (L: T:P): 3:0:0	Credits: 1
Type of course: Lecture + Assignments	Total Contact Hours:12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of ethics, values, and the role of human behavior in society.

Course Objectives:

1. Develop an understanding of human values and ethical principles.
2. Analyze the significance of values and ethics in personal and professional life.
3. Explore ethical decision-making and its impact on society.
4. Study the role of human values in fostering social responsibility and justice.
5. Examine the relationship between moral values and cultural diversity.

Course Content:

Topic and Contents
BLOCK-1: Introduction to Human Values
Unit 1: Definition and Importance of Human Values Unit 2: Types of Human Values (Moral, Social, Religious, Cultural) Unit 3: Role of Values in Personal Development
BLOCK-2: Ethical Theories and Principles
Unit 4: Major Ethical Theories (Utilitarianism, Deontology, Virtue Ethics) Unit 5: Moral Dilemmas and Ethical Decision-Making Unit 6: Ethical Standards and Codes of Conduct
BLOCK-3: Values in Professional Life
Unit 7: Ethical Issues in Business and Professionalism Unit 8: Social Responsibility and Corporate Ethics Unit 9: Ethics in Technology and Innovation

BLOCK-4: Ethics in Social Context
Unit 10: Ethics in Governance and Law Unit 11: Environmental Ethics and Sustainability Unit 12: Human Rights and Social Justice
BLOCK-5: Developing Ethical Leadership
Unit 13: Characteristics of Ethical Leaders Unit 14: Ethical Leadership in Organizations Unit 15: Promoting Ethical Behavior in Society

Course Outcomes:

1. Understand the fundamental concepts of human values and ethics.
2. Critically evaluate ethical dilemmas and decision-making processes.
3. Apply ethical principles in professional and social settings.
4. Foster social responsibility and a commitment to justice.
5. Develop leadership skills based on ethical principles.

Reference:

1. Rachels, J., & Rachels, S. (2019). The elements of moral philosophy (9th ed.). McGraw-Hill Education.
2. Shaw, W. H. (2016). Business ethics: A textbook with cases (9th ed.). Cengage Learning.
3. Velasquez, M. G. (2017). Business ethics: Concepts and cases (8th ed.). Pearson Education.
4. Beauchamp, T. L., & Childress, J. F. (2019). Principles of biomedical ethics (8th ed.). Oxford University Press.
5. Freeman, R. E., Harrison, J., & Freeman, A. (2018). Stakeholder theory: The state of the art. Cambridge University Press.

Program: Bachelor of Computer Applications

Course Name: Environmental Studies	Course Code: DCA112
Semester: 2	Core / Elective: Core
Teaching Scheme in Hrs (L: T:P): 3:0:0	Credits: 3
Type of course: Lecture + Assignments	Total Contact Hours:12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of the environment and its role in human life.

Course Objectives:

1. Understand the basic principles of environmental science and its relevance.
2. Study the impact of human activities on the environment.
3. Learn about various natural resources, conservation techniques, and sustainability practices.
4. Gain knowledge of environmental policies and their implications.
5. Develop the skills to promote environmental awareness and sustainability.

Course Content:

Topic and Contents
BLOCK-1: Introduction to Environmental Studies
Unit 1: Definition, Scope, and Importance of Environmental Studies Unit 2: Ecosystems and their Components Unit 3: Types of Ecosystems and Energy Flow
BLOCK-2: Natural Resources
Unit 4: Types of Natural Resources (Renewable and Non-Renewable) Unit 5: Resource Depletion and Conservation Strategies Unit 6: Sustainable Management of Natural Resources
BLOCK-3: Environmental Pollution
Unit 7: Air, Water, and Soil Pollution Unit 8: Causes, Effects, and Control of Pollution Unit 9: Waste Management and Recycling

BLOCK-4: Environmental Protection and Conservation
Unit 10: Biodiversity and Conservation
Unit 11: Environmental Legislation and Policies
Unit 12: Role of NGOs and International Organizations in Environmental Protection
BLOCK-5: Climate Change and Global Environmental Issues
Unit 13: Global Warming and Climate Change
Unit 14: Environmental Issues and Sustainable Development Goals (SDGs)
Unit 15: Green Technologies and Renewable Energy Solutions

Course Outcomes:

1. Understand the environmental challenges facing the planet today.
2. Learn to assess human activities and their effects on the environment.
3. Apply the concepts of conservation and sustainable practices in everyday life.
4. Familiarize with global environmental policies and sustainability frameworks.
5. Engage in environmental advocacy and contribute to solving environmental issues

Reference:

1. Cunningham, W. P., & Cunningham, M. A. (2017). *Principles of environmental science: Inquiry and applications* (9th ed.). McGraw-Hill Education.
2. Miller, G. T., & Spoolman, S. E. (2019). *Living in the environment* (19th ed.). Cengage Learning.
3. Raven, P. H., Berg, L. R., & Hassenzahl, D. M. (2020). *Environment* (11th ed.). Wiley.
4. Carson, R. (2002). *Silent spring*. Houghton Mifflin Harcourt.
5. Kormondy, E. J., & Brown, D. S. (2013). *Environmental science: A global concern* (12th ed.). McGraw-Hill Education.

SYLLABUS

(SEMESTER-III)

Program: Bachelor of Computer Applications

Course Name: Financial Management	Course Code: DCA201
Semester: 3	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 4
Type of course: Lecture + Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of accounting and financial principles.

Course Objectives:

1. Develop an understanding of financial management concepts and their application in business.
2. Learn the tools and techniques used in financial planning, control, and analysis.
3. Analyze the role of financial management in decision-making and strategy formulation.
4. Gain knowledge of capital budgeting, risk management, and financing decisions.
5. Understand the financial markets and institutions and their role in the economy.

Course Content:

Topic and Contents
BLOCK-1: Introduction to Financial Management
Unit 1: Definition, Scope, and Objectives of Financial Management Unit 2: Financial Planning and Control Unit 3: Financial Statements and Analysis
BLOCK-2: Time Value of Money
Unit 4: Concept of Time Value of Money Unit 5: Compound Interest and Discounting Unit 6: Present Value and Future Value Calculations
BLOCK-3: Capital Budgeting
Unit 7: Methods of Capital Budgeting (NPV, IRR, Payback Period) Unit 8: Risk and Uncertainty in Capital Budgeting Unit 9: Cost of Capital

BLOCK-4: Financial Markets and Institutions
Unit 10: Structure and Functioning of Financial Markets Unit 11: Types of Financial Institutions and their Role Unit 12: Financial Instruments and Derivatives
BLOCK-5: Working Capital and Dividend Decisions
Unit 13: Working Capital Management Unit 14: Dividend Policy and its Impact on Firm Value Unit 15: Short-term Financing and Liquidity Management

Course Outcomes:

1. Develop a clear understanding of financial management principles and tools.
2. Apply financial analysis techniques to assess business performance.
3. Make informed decisions regarding capital budgeting and financial planning.
4. Understand the role of financial markets in the economy and business operations.
5. Evaluate dividend policies and working capital strategies to optimize firm performance.

Reference:

1. Brigham, E. F., & Ehrhardt, M. C. (2017). Financial management: Theory and practice (15th ed.). Cengage Learning.
2. Ross, S. A., Westerfield, R. W., & Jordan, B. D. (2019). Fundamentals of corporate finance (12th ed.). McGraw-Hill Education.
3. Van Horne, J. C., & Wachowicz, J. M. (2008). Fundamentals of financial management (13th ed.). Pearson Education.
4. Khan, M. Y., & Jain, P. K. (2015). Financial management (8th ed.). Tata McGraw-Hill Education.
5. Gitman, L. J. (2015). Principles of managerial finance (14th ed.). Pearson Education.

Program: Bachelor of Computer Applications

Course Name: Database Management System (DBMS)	Course Code: DCA203
Semester: 3	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 4
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of computer systems and programming fundamentals.

Course Objectives:

1. Understand the fundamental concepts of database management systems and their architecture.
2. Learn the principles of data modeling using Entity-Relationship (ER) and relational models.
3. Master SQL for querying and managing relational databases.
4. Explore concepts like normalization, indexing, and transactions for efficient database design.
5. Gain insights into advanced topics such as distributed databases and NoSQL systems.

Course Content:

Topic and Contents
BLOCK-1: Introduction to DBMS
Unit 1: Overview of Database Systems Unit 2: DBMS Architecture and Data Models Unit 3: Data Independence and Database Applications
BLOCK-2: Relational Model and SQL
Unit 4: Relational Model Concepts Unit 5: Basics of SQL (DDL, DML, and DCL) Unit 6: Advanced SQL Queries (Joins, Views, and Sub queries)

BLOCK-3: Database Design
Unit 7: Entity-Relationship (ER) Modeling Unit 8: Relational Database Design and Normalization Unit 9: Functional Dependencies and Decomposition
BLOCK-4: Transaction Management and Indexing
Unit 10: Transactions and ACID Properties Unit 11: Concurrency Control and Recovery Mechanisms Unit 12: Indexing and Hashing Techniques
BLOCK-5: Advanced Topics in DBMS
Unit 13: Distributed and Parallel Databases Unit 14: Introduction to No SQL Databases Unit 15: Big Data and Cloud-Based Database Systems

Course Outcomes:

1. Explain the core concepts of DBMS and its applications.
2. Design efficient database schemas using ER modeling and normalization techniques.
3. Write and execute complex SQL queries for database management.
4. Analyze transaction management and concurrency control mechanisms.
5. Explore advanced database technologies, including distributed and NoSQL systems.

Reference:

1. Korth, H. F., Silberschatz, A., & Sudarshan, S. (2019). Database system concepts (7th ed.). McGraw-Hill Education.
2. Elmasri, R., & Navathe, S. B. (2015). Fundamentals of database systems (7th ed.). Pearson Education.
3. Date, C. J. (2004). An introduction to database systems (8th ed.). Addison-Wesley.
4. Ramakrishnan, R., & Gehrke, J. (2014). Database management systems (3rd ed.). McGraw-Hill Education.
5. Connolly, T. M., & Begg, C. E. (2014). Database systems: A practical approach to design, implementation, and management (6th ed.). Pearson Education.

Program: Bachelor of Computer Applications

Course Name: Web Architecture Design	Course Code: DCA205
Semester: 3	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 4
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic knowledge of computer systems and familiarity with programming fundamentals.

Course Objectives:

1. Understand the principles of web technologies and their role in modern applications.
2. Learn HTML, CSS, and JavaScript for designing and developing interactive web pages.
3. Explore backend development using server-side technologies and databases.
4. Gain proficiency in responsive design and frameworks for better user experiences.
5. Develop a full-stack web application as a project.

Course Content:

Topic and Contents
BLOCK-1: Introduction to Web Development
Unit 1: Basics of the Web: HTTP, URLs, and Web Servers Unit 2: Introduction to HTML5: Tags, Elements, and Structure Unit 3: Basics of CSS3: Styling and Layouts
BLOCK-2: Client-Side Scripting
Unit 4: Introduction to JavaScript: Syntax and DOM Manipulation Unit 5: Event Handling and Form Validation Unit 6: Introduction to Frontend Frameworks (e.g., Bootstrap)

BLOCK-3: Backend Development
Unit 7: Basics of Server-Side Scripting (e.g., PHP/Node.js)
Unit 8: Database Integration (SQL/NoSQL)
Unit 9: REST ful APIs and Server Communication
BLOCK-4: Advanced Web Development
Unit 10: Responsive Web Design and Media Queries
Unit 11: Introduction to SPA Frameworks (e.g., React/Angular Basics)
Unit 12: Web Hosting and Deployment Strategies
BLOCK-5: Security and Project Development
Unit 13: Web Security Basics (XSS, CSRF, and HTTPS)
Unit 14: Performance Optimization for Web Applications
Unit 15: Capstone Project: Develop a Full-Stack Web Application

Course Outcomes:

1. Build structured and responsive web pages using HTML and CSS.
2. Develop interactive client-side applications using JavaScript.
3. Implement server-side logic and database interactions.
4. Apply best practices in web security and performance optimization.
5. Deploy a complete web application demonstrating full-stack development skills.

Reference:

1. Richards, M., & Ford, P. (2021). Software architecture for developers (2nd ed.). Lean Publishing.
2. Fowler, M. (2003). Patterns of enterprise application architecture. Addison-Wesley.
3. Shklar, L., & Hildebrandt, R. (2007). Web application architecture: Principles, protocols, and practices. Wiley.
4. Mowbray, T. J., & Malveau, R. C. (2003). Software architecture in practice (2nd ed.). Addison-Wesley.
5. Fielding, R. T. (2000). Architectural styles and the design of network-based software architectures (Doctoral dissertation, University of California, Irvine).

Program: Bachelor of Computer Applications

Course Name: Web Architecture Design Lab	Course Code: DCA251
Semester: 3	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 1
Type of course: LAB	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of HTML, CSS, Java Script, and web design principles.

Course Objectives:

1. Provide hands-on experience with front-end technologies such as HTML, CSS, and JavaScript.
2. Implement interactive web pages using JavaScript and DOM manipulation.
3. Understand and work with server-side technologies and databases for dynamic websites.
4. Develop responsive and optimized web pages for different screen sizes and devices.
5. Apply web security principles and learn deployment strategies for live websites.

Course Content:

Topic and Contents
Lab 1: Design a static web page using HTML5 and CSS3, including text formatting, images, links, and tables
Lab 2: Create a form with input validation using JavaScript (e.g., email validation, required fields).
Lab 3: Implement navigation menus and interactive elements using JavaScript (e.g., dropdowns, modals).
Lab 4: Use CSS Grid and Flex box to design a responsive webpage layout.
Lab 5: Develop a dynamic web page that interacts with user input and changes content using JavaScript and DOM.
Lab 6: Create a simple CRUD (Create, Read, Update, Delete) application using PHP or Node.js and connect to a My SQL or No SQL database.

Lab 7: Work with external APIs to fetch and display data (e.g., weather data or user information).
Lab 8: Implement AJAX for creating dynamic and asynchronous web applications.
Lab 9: Develop a single-page application (SPA) using a front-end framework like React or Angular.
Lab 10: Deploy a web application to a hosting platform (e.g., Git Hub Pages, Net lify, or Heroku) and ensure proper configuration for live hosting

Course Outcomes:

1. Design and develop static and dynamic web pages using HTML, CSS, and JavaScript.
2. Create responsive and mobile-friendly layouts using modern web design techniques.
3. Develop web applications that interact with databases and handle user input.
4. Understand and apply web security best practices for protecting web applications.
5. Deploy and maintain web applications on various hosting platforms.

Reference:

1. Richards, M., & Ford, P. (2021). Software architecture for developers (2nd ed.). Lean Publishing.
2. Fowler, M. (2003). Patterns of enterprise application architecture. Addison-Wesley.
3. Shklar, L., & Hildebrandt, R. (2007). Web application architecture: Principles, protocols, and practices. Wiley.
4. Mowbray, T. J., & Malveau, R. C. (2003). Software architecture in practice (2nd ed.). Addison-Wesley.
5. Fielding, R. T. (2000). Architectural styles and the design of network-based software architectures (Doctoral dissertation, University of California, Irvine).

Program: Bachelor of Computer Applications

Course Name: Data Visualization Techniques	Course Code: DCA207
Semester: 3	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 4
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of data analysis and statistics.

Course Objectives:

1. Introduce the fundamental concepts of data visualization and its importance in data analysis.
2. Teach various techniques and tools for effective data visualization.
3. Understand how to choose appropriate visualizations for different types of data.
4. Explore the use of advanced visualization techniques and interactive dashboards.
5. Develop the ability to communicate complex data insights clearly and effectively using visualizations.

Course Content:

Topic and Contents
BLOCK-1: Introduction to Data Visualization
Unit 1: Introduction to Data Visualization Unit 2: Types of Data and their Representation Unit 3: Principles of Effective Data Visualization
BLOCK-2: Basic Data Visualization Techniques
Unit 4: Bar Charts and Histograms Unit 5: Line and Area Charts Unit 6: Pie Charts and Donut Charts

BLOCK-3: Advanced Visualization Techniques
Unit 7: Scatter Plots and Bubble Charts Unit 8: Heat maps and Tree Maps Unit 9: Box Plots and Violin Plots
BLOCK-4: Interactive Data Visualizations
Unit 10: Transactions and ACID Properties Unit 11: Concurrency Control and Recovery Mechanisms Unit 12: Indexing and Hashing Techniques
BLOCK-5: Data Storytelling and Visualization Design
Unit 13: Data Storytelling Techniques Unit 14: Visual Perception and Color Theory in Data Visualization Unit 15: Best Practices for Designing Data Visualizations

Course Outcomes:

1. Understand the fundamental principles of data visualization and its role in data analysis.
2. Learn to choose and apply appropriate visualization techniques for different types of data.
3. Create clear and effective visualizations that convey data insights.
4. Use advanced tools to create interactive and dynamic visualizations.
5. Design data visualizations that are aesthetically pleasing and accessible

Reference:

1. Few, S. (2012). Show me the numbers: Designing tables and graphs to enlighten (2nd ed.). Analytics Press.
2. Tufte, E. R. (2001). The visual display of quantitative information (2nd ed.). Graphics Press.
3. Kirk, A. (2016). Data visualization: A handbook for data driven design. Sage Publications.
4. Ware, C. (2013). Information visualization: Perception for design (3rd ed.). Morgan Kaufmann.
5. Cairo, A. (2012). The functional art: An introduction to information graphics and visualization. New Riders.

Program: Bachelor of Computer Applications

Course Name: Data Visualization Techniques Lab	CourseCode:DCA253
Semester: 3	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 1
Type of course: LAB	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of data analysis and statistics.

Course Objectives:

1. Develop practical skills in creating a variety of data visualizations.
2. Gain hands-on experience with popular data visualization tools like Tableau, Power BI, and D3.js.
3. Learn to interpret and present data insights using interactive and static visualizations.
4. Understand how to design effective visualizations for real-world data.
5. Apply visualization techniques to create dashboards and reports that tell compelling data stories.

Course Content:

Topic and Contents
Lab 1: Introduction to Data Visualization Tools
Lab 2: Creating Static Visualizations
Lab 3: Advanced Visualizations
Lab 4: Interactive Data Visualization
Lab 5: Data Storytelling and Design
Lab 6: Real-World Data Project
Lab 7: Advanced Custom Visualizations (D3.js)
Lab 8: Visualization for Business Intelligence
Lab 9: Data Visualization with Geospatial Data
Lab 10: Final Project

Course Outcomes:

1. Gain hands-on experience with popular data visualization software tools.
2. Create effective and visually appealing charts and graphs.
3. Develop interactive visualizations and dashboards.
4. Apply advanced data visualization techniques to real-world datasets.
5. Build and present data-driven insights that tell a compelling story.

Reference:

1. Few, S. (2012). Show me the numbers: Designing tables and graphs to enlighten (2nd ed.). Analytics Press.
2. Tufte, E. R. (2001). The visual display of quantitative information (2nd ed.). Graphics Press.
3. Kirk, A. (2016). Data visualization: A handbook for data driven design. Sage Publications.
4. Ware, C. (2013). Information visualization: Perception for design (3rd ed.). Morgan Kaufmann.
5. Cairo, A. (2012). The functional art: An introduction to information graphics and visualization. New Riders.

Program: Bachelor of Computer Applications

Course Name: Programming in Java	Course Code: DCA209
Semester: 3	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 4
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic knowledge of computer science concepts and problem-solving skills.

Course Objectives:

1. Develop an understanding of Java programming language syntax and structure.
2. Learn object-oriented programming (OOP) concepts such as inheritance, polymorphism, and encapsulation.
3. Implement control flow, data structures, and algorithms using Java.
4. Understand and apply Java libraries and frameworks for various programming tasks.
5. Build and debug Java applications effectively.

Course Content:

Topic and Contents
BLOCK-1: Introduction to Java Programming
Unit 1: Overview of Java Programming Language Unit 2: Java Development Environment and Setup Unit 3: Basic Syntax, Variables, and Data Types
BLOCK-2: Control Structures and Arrays
Unit 4: Operators, Expressions, and Control Statements Unit 5: Arrays and Multi-dimensional Arrays Unit 6: String Manipulation and Handling
BLOCK-3: Object-Oriented Programming in Java
Unit 7: Classes and Objects Unit 8: Constructors, Methods, and Instance Variables Unit 9: Inheritance and Polymorphism

BLOCK-4: Exception Handling and File I/O
Unit 10: Exception Handling (Try, Catch, Finally)
Unit 11: File Handling and Streams
Unit 12: Working with Directories and File I/O Operations
BLOCK-5: Java Collections Framework and GUI
Unit 13: Introduction to Java Collections (List, Set, Map)
Unit 14: Java Generics and Collection Operations
Unit 15: Introduction to Java GUI with Swing

Course Outcomes:

1. Gain a solid foundation in Java programming language syntax and structure.
2. Master object-oriented programming concepts and implement them in Java.
3. Solve complex problems using control structures, arrays, and data types.
4. Handle exceptions and work with file I/O operations in Java.
5. Develop basic graphical user interfaces (GUIs) using Java Swing.

Reference:

1. Bloch, J. (2018). Effective Java (3rd ed.). Addison-Wesley.
2. Horstmann, C. S. (2019). Core Java Volume I--Fundamentals (11th ed.). Pearson Education.
3. Schildt, H. (2017). Java: The complete reference (10th ed.). McGraw-Hill Education.
4. Eckel, B. (2006). Thinking in Java (4th ed.). Prentice Hall.
5. Sierra, K., & Bates, B. (2005). Head first Java (2nd ed.). O'Reilly Media.

Program: Bachelor of Computer Applications

Course Name: Programming in Java Lab	Course Code: DCA255
Semester: 3	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 1
Type of course: LAB	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic knowledge of computer science concepts and familiarity with Java programming fundamentals.

Course Objectives:

1. Develop hands-on experience with Java programming concepts and syntax.
2. Implement object-oriented programming principles in Java.
3. Work with control structures, arrays, and data types in Java applications.
4. Apply exception handling, file I/O, and Java Collections in real-world problems.
5. Build simple Java GUI applications using Java Swing.

Course Content:

Topic and Contents
Lab 1: Write a simple Java program that outputs "Hello, World!"
Lab 2: Develop a program to find the largest number in an array.
Lab 3: Write a program to calculate the factorial of a number using recursion.
Lab 4: Develop a program to reverse a string without using built-in functions.
Lab 5: Implement a simple class to model a Book object with attributes like title, author, and price.
Lab 6: Write a program to check whether a given number is prime or not.
Lab 7: Write a program to implement the basic operations of a calculator (addition, subtraction, multiplication, division).
Lab 8: Create a Java program to handle file input/output (I/O) operations, such as reading from and writing to a file.
Lab 9: Create a simple program to implement a student class and display student details.
Lab 10: Build a simple Java GUI application using Swing to input and display user data (e.g., name, age, email).

Course Outcomes:

1. Write, debug, and run Java programs using various programming constructs.
2. Apply object-oriented principles such as inheritance and polymorphism in Java.
3. Manage and manipulate arrays and collections in Java effectively.
4. Handle exceptions and perform file I/O operations in Java applications.
5. Design and implement simple Java GUI applications using Java Swing.

Reference:

1. Bloch, J. (2018). Effective Java (3rd ed.). Addison-Wesley.
2. Horstmann, C. S. (2019). Core Java Volume I--Fundamentals (11th ed.). Pearson Education.
3. Schildt, H. (2017). Java: The complete reference (10th ed.). McGraw-Hill Education.
4. Eckel, B. (2006). Thinking in Java (4th ed.). Prentice Hall.
5. Sierra, K., & Bates, B. (2005). Head first Java (2nd ed.). O'Reilly Media.

SYLLABUS

(SEMESTER-IV)

Program: Bachelor of Computer Applications

Course Name: Computer Organization & Architecture	Course Code: DCA202
Semester: 4	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 4
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of computer systems and digital electronics.

Course Objectives:

1. Understand the basic structure and operation of a computer system.
2. Learn about the internal components and their interactions in a computer.
3. Study data representation and instruction set architectures.
4. Understand memory hierarchy and CPU organization.
5. Learn about input-output systems and parallel processing.

Course Content:

Topic and Contents
BLOCK-1: Introduction to Computer Organization
Unit 1: Basics of Computer Organization Unit 2: Computer System Architecture Unit 3: Von Neumann Architecture
BLOCK-2: Data Representation
Unit 4: Number Systems and Binary Arithmetic Unit 5: Character Representation (ASCII, Unicode) Unit 6: Floating Point Representation
BLOCK-3: CPU Organization
Unit 7: CPU Architecture and Functions Unit 8: ALU and Control Unit Unit 9: Instruction Cycle and Execution

BLOCK-4: Memory Organization
Unit 10: Types of Memory (Primary and Secondary)
Unit 11: Memory Hierarchy and Cache Memory
Unit 12: Virtual Memory and Paging
BLOCK-5: Input/ Output Systems and Parallel Processing
Unit 13: Input/ Output Devices and Interfaces
Unit 14: Interrupts and Direct Memory Access (DMA)
Unit 15: Parallel Processing and Pipelining

Course Outcomes:

1. Understand the architecture and design principles of computer systems.
2. Develop knowledge of the components involved in CPU and memory operations.
3. Apply the concepts of data representation in solving computer problems.
4. Evaluate the functioning of input/output devices and their integration with computer systems.
5. Gain an understanding of modern processor architecture, parallelism, and pipelining techniques.

Reference:

1. Stallings, W. (2021). Computer organization and architecture: Designing for performance (11th ed.). Pearson Education.
2. Hamacher, C. V., Vranesic, Z. G., Zaky, S. G., & Manjikian, N. (2012). Computer organization and embedded systems (6th ed.). McGraw-Hill Education.
3. Patterson, D. A., & Hennessy, J. L. (2021). Computer organization and design: The hardware/software interface (6th ed.). Morgan Kaufmann.
4. Mano, M. M., & Ciletti, M. D. (2017). Digital design: With an introduction to the Verilog HDL, VHDL, and SystemVerilog (6th ed.). Pearson Education.
5. Heuring, V. P., & Jordan, H. F. (2003). Computer systems design and architecture (2nd ed.). Pearson Education.

Program: Bachelor of Computer Applications

Course Name: Software Engineering Fundamentals	Course Code: DCA204
Semester: 4	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 4
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of programming, algorithms, and data structures.

Course Objectives:

1. Understand the software development lifecycle and methodologies.
2. Learn how to design software systems using appropriate tools and techniques.
3. Study best practices for software testing, quality assurance, and maintenance.
4. Gain knowledge of project management and team collaboration in software projects.
5. Develop skills to create scalable, maintainable, and user-friendly software.

Course Content:

Topic and Contents
BLOCK-1: Introduction to Software Engineering
Unit 1: Software Engineering Concepts Unit 2: Software Development Models (Waterfall, Agile, etc.) Unit 3: Software Process Frameworks and Methodologies
BLOCK-2: Software Requirements and Design
Unit 4: Requirements Engineering: Gathering and Documentation Unit 5: System Design: High-Level and Low-Level Design Unit 6: UML Diagrams for System Modeling
BLOCK-3: Software Development
Unit 7: Programming Languages and Tools for Software Development Unit 8: Coding Standards and Best Practices Unit 9: Version Control Systems and Software Configuration Management

BLOCK-4: Software Testing and Quality Assurance
Unit 10: Types of Software Testing (Unit, Integration, System, and Acceptance) Unit 11: Automated Testing and Test-Driven Development (TDD) Unit 12: Software Metrics and Quality Assurance Techniques
BLOCK-5: Software Maintenance and Project Management
Unit 13: Software Maintenance and Evolution Unit 14: Risk Management in Software Projects Unit 15: Project Management and Team Collaboration Tools (JIRA, Git, etc.)

Course Outcomes:

1. Explain software engineering principles and the software development process.
2. Design and develop software systems by analyzing requirements and applying appropriate methodologies.
3. Implement software testing techniques to ensure the quality and correctness of the software.
4. Understand the importance of software maintenance and effectively manage evolving systems.
5. Use project management tools and techniques to manage software projects and collaborate within teams.

Reference:

1. Sommerville, I. (2015). Software engineering (10th ed.). Pearson Education.
2. Pressman, R. S., & Maxim, B. R. (2019). Software engineering: A practitioner's approach (9th ed.). McGraw-Hill Education.
3. Fairley, R. E. (2011). Managing and leading software projects. Wiley.
4. Boehm, B. W. (1981). Software engineering economics. Prentice Hall.
5. Jalote, P. (2005). An integrated approach to software engineering (3rd ed.). Springer.

Program: Bachelor of Computer Applications

Course Name: Advanced Web Development	Course Code: DCA206
Semester: 4	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 4
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic knowledge of programming and HTML.

Course Objectives:

1. Understand the fundamentals of PHP and its integration with web technologies.
2. Learn to build dynamic and interactive web pages.
3. Explore server-side scripting and database integration using PHP.
4. Understand error handling and debugging in PHP applications.
5. Gain hands-on experience with common PHP frameworks and tools

Course Content:

Topic and Contents
BLOCK-1: Introduction to PHP
Unit 1: Basics of PHP: Syntax, Variables, and Data Types Unit 2: Control Structures: Conditionals and Loops Unit 3: Functions in PHP
BLOCK-2: Working with Forms and Sessions
Unit 4: Handling HTML Forms with PHP Unit 5: Cookies and Sessions Management Unit 6: File Handling in PHP
BLOCK-3: Database Connectivity
Unit 7: Introduction to My SQL and PHP Integration Unit 8: CRUD Operations with My SQL Unit 9: Prepared Statements and Security in PHP

BLOCK-4: Advanced PHP Concepts
Unit 10: Object-Oriented Programming in PHP Unit 11: Error Handling and Exception Management Unit 12: PHP Frameworks Overview (Laravel/ Code Igniter)
BLOCK-5: PHP in Real-World Applications
Unit 13: Building a Blog or CMS with PHP Unit 14: Integrating APIs in PHP Applications Unit 15: Deployment and Optimization of PHP Projects

Course Outcomes:

1. Develop dynamic websites using PHP and My SQL.
2. Manage form data securely and handle user sessions effectively.
3. Perform CRUD operations and ensure secure database connectivity.
4. Apply OOP principles and error handling in PHP projects.
5. Design and deploy PHP-based web applications with real-world functionality.

Reference:

1. Duckett, J. (2014). JavaScript and JQuery: Interactive front-end web development. Wiley.
2. Flanagan, D. (2020). JavaScript: The definitive guide: Master the world's most-used programming language (7th ed.). O'Reilly Media.
3. Hartl, M. (2022). Ruby on Rails tutorial: Learn web development with Rails (7th ed.). Addison-Wesley.
4. Keith, J. (2020). HTML5 for web designers (2nd ed.). A Book Apart.
5. Wieruch, R. (2021). The road to React: Your journey to master React.js in JavaScript. Independently Published.

Program: Bachelor of Computer Applications

Course Name: Advanced Web Development Lab	Course Code: DCA252
Semester: 4	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 1
Type of course: LAB	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of programming and web development concepts.

Course Objectives:

1. Gain practical experience in PHP programming and web application development.
2. Learn to handle forms, process user inputs, and maintain sessions.
3. Understand database connectivity using PHP and My SQL.
4. Implement dynamic and interactive features in web applications.
5. Debug and optimize PHP applications for better performance.

Course Content:

Topic and Contents
Lab 1: Write a program to display "Hello World" and demonstrate the use of variables and data types.
Lab 2: Create a PHP program using conditional statements to check if a number is odd or even.
Lab 3: Develop a PHP program using loops to generate the multiplication table of a number.
Lab 4: Create a form in HTML and process its data using PHP (e.g., a user registration form).
Lab 5: Implement session handling to track user login and logout processes.
Lab 6: Write a PHP program to read and write data to a file (e.g., saving user feedback).
Lab 7: Connect PHP with My SQL and perform basic CRUD operations (e.g., managing student records).
Lab 8: Build a PHP program to display dynamic content based on user selection (e.g., product details).
Lab 9: Validate user inputs in a form and prevent SQL injection attacks.
Lab 10: Design a mini-project to develop a basic CMS or a blog using PHP and MySQL.

Course Outcomes:

1. Apply PHP programming to develop dynamic web pages.
2. Demonstrate the ability to handle user inputs and manage sessions.
3. Integrate PHP with My SQL for database-driven applications.
4. Ensure secure and validated inputs in PHP applications.
5. Create functional web applications showcasing learned concepts.

Reference:

1. Duckett, J. (2014). JavaScript and JQuery: Interactive front-end web development. Wiley.
2. Flanagan, D. (2020). JavaScript: The definitive guide: Master the world's most-used programming language (7th ed.). O'Reilly Media.
3. Hartl, M. (2022). Ruby on Rails tutorial: Learn web development with Rails (7th ed.). Addison-Wesley.
4. Keith, J. (2020). HTML5 for web designers (2nd ed.). A Book Apart.
5. Wieruch, R. (2021). The road to React: Your journey to master React.js in JavaScript. Independently Published.

Program: Bachelor of Computer Applications

Course Name: Data Structures and Algorithms (DSA)	Course Code: DCA208
Semester: 4	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits:4
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE:70 Marks

Pre-requisites:

Basic understanding of programming concepts and logical problem-solving skills.

Course Objectives:

1. Learn fundamental concepts of data structures and their applications in problem-solving.
2. Understand algorithm design techniques for optimized solutions.
3. Analyze time and space complexity of algorithms.
4. Develop skills to implement data structures in programming languages.
5. Solve real-world problems using appropriate data structures and algorithms.

Course Content:

Topic and Contents
BLOCK-1: Introduction to Data Structures and Algorithms
Unit 1: Overview of Data Structures and Algorithms Unit 2: Time and Space Complexity Analysis Unit 3: Recursion and Divide-and-Conquer Approach
BLOCK-2: Linear Data Structures
Unit 4: Arrays and Linked Lists Unit 5: Stacks and Queues Unit 6: Applications of Stacks and Queues
BLOCK-3: Non-Linear Data Structures
Unit 7: Trees and Binary Trees Unit 8: Binary Search Trees (BST) and Heaps Unit 9: Graphs: Representation and Traversal (BFS, DFS)

BLOCK-4: Searching and Sorting Algorithms
Unit 10: Searching Algorithms (Linear, Binary Search)
Unit 11: Sorting Algorithms (Bubble, Merge, Quick, and Heap Sort)
Unit 12: Hashing Techniques and Applications
BLOCK-5: Advanced Topics and Applications
Unit 13: Greedy Algorithms and Dynamic Programming
Unit 14: Backtracking and Branch-and-Bound Techniques
Unit 15: String Matching Algorithms and Applications

Course Outcomes:

1. Understand and implement fundamental data structures such as arrays, linked lists, stacks, and queues.
2. Develop and analyze algorithms for searching, sorting, and traversal in linear and non-linear data structures.
3. Apply advanced techniques like greedy algorithms and dynamic programming for optimized problem-solving.
4. Implement and evaluate real-world applications using graphs, trees, and hashing.
5. Gain proficiency in algorithmic problem-solving for competitive programming and software development.

Reference:

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms (3rd ed.). MIT Press.
2. Sahni, S. (2005). Data structures, algorithms, and applications in C++ (2nd ed.). Universities Press.
3. Weiss, M. A. (2014). Data structures and algorithm analysis in C++ (4th ed.). Pearson Education.
4. Lafore, R. (2002). Data structures and algorithms in Java. Sams Publishing.
5. Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2014). Data structures and algorithms in Python. Wiley.

Program: Bachelor of Computer Applications

Course Name: Data Structures and Algorithms (DSA) Lab	Course Code: DCA254
Semester: 4	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 1
Type of course: LAB	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic knowledge of programming in C, C++, or Python.

Course Objectives:

1. Provide hands-on experience in implementing data structures and algorithms.
2. Develop problem-solving skills using recursive and iterative techniques.
3. Familiarize students with various real-world applications of data structures.
4. Enhance debugging and testing skills for algorithmic programs.
5. Improve efficiency in code writing by applying appropriate algorithms.

Course Content:

Topic and Contents
Lab 1: Implement array operations (insertion, deletion, traversal, and searching).
Lab 2: Create singly, doubly, and circular linked lists with basic operations.
Lab 3: Implement stack operations using arrays and linked lists, and solve a real-world problem (e.g., expression evaluation).
Lab 4: Implement queue and circular queue operations using arrays and linked lists.
Lab 5: Develop programs for tree traversal methods (inorder, preorder, postorder) and binary search trees.
Lab 6: Implement graph traversal techniques (BFS and DFS) and detect cycles in a graph.
Lab 7: Write programs for sorting algorithms (bubble, merge, quick, and heap sort).
Lab 8: Implement hashing with open addressing and chaining methods for collision resolution.
Lab 9: Solve problems using dynamic programming (e.g., longest common subsequence, knapsack problem).
Lab 10: Implement string matching algorithms (e.g., Rabin-Karp, KMP algorithm).

Course Outcomes:

1. Implement and test various linear and non-linear data structures.
2. Apply sorting and searching algorithms to solve real-world problems.
3. Solve advanced problems using graphs, trees, and dynamic programming.
4. Develop efficient algorithms with minimal time and space complexity.
5. Enhance proficiency in implementing algorithmic solutions for competitive programming.

Reference:

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms (3rd ed.). MIT Press.
2. Sahni, S. (2005). Data structures, algorithms, and applications in C++ (2nd ed.). Universities Press.
3. Weiss, M. A. (2014). Data structures and algorithm analysis in C++ (4th ed.). Pearson Education.
4. Lafore, R. (2002). Data structures and algorithms in Java. Sams Publishing.
5. Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2014). Data structures and algorithms in Python. Wiley.

Program: Bachelor of Computer Applications

Course Name: Data Warehousing and Data Mining	Course Code: DCA210
Semester: 4	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 4
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of databases and SQL.

Course Objectives:

1. Understand the concepts and architecture of data warehousing.
2. Learn ETL processes and multidimensional data modeling techniques.
3. Gain knowledge of OLAP operations and data warehouse design.
4. Explore data mining techniques and algorithms. intelligence.
5. Apply data warehousing and mining methods for decision-making and business

Course Content:

Topic and Contents
BLOCK-1: Introduction to Data Warehousing
Unit 1: Overview of Data Warehousing Unit 2: Data Warehouse Architecture and Components Unit 3: ETL Processes and Data Integration
BLOCK-2: Data Warehouse Design
Unit 4: Multidimensional Data Modeling (Star and Snowflake Schemas) Unit 5: OLAP Operations (Roll-up, Drill-down, Slice, Dice, and Pivot) Unit 6: Data Warehouse Implementation and Challenges
BLOCK-3: Introduction to Data Mining
Unit 7: Data Mining Concepts and Applications Unit 8: Data Preprocessing Techniques Unit 9: Data Mining Process and CRISP-DM Model

BLOCK-4: Data Mining Techniques and Algorithms
Unit 10: Classification Techniques (Decision Trees, Naïve Bayes)
Unit 11: Clustering Techniques (K-Means, Hierarchical Clustering)
Unit 12: Association Rule Mining (Apriori and FP-Growth)
BLOCK-5: Advanced Topics and Applications
Unit 13: Web Mining and Text Mining
Unit 14: Big Data Analytics and Mining Tools
Unit 15: Data Warehousing and Mining for Business Intelligence

Course Outcomes:

1. Explain the architecture and operations of a data warehouse.
2. Design multidimensional data models and implement ETL processes.
3. Apply OLAP techniques for complex business queries.
4. Use classification, clustering, and association algorithms for data mining.
5. Leverage data warehousing and mining for real-world decision-making applications.

Reference:

1. Han, J., Pei, J., & Kamber, M. (2011). Data mining: Concepts and techniques (3rd ed.). Elsevier.
2. Inmon, W. H. (2005). Building the data warehouse (4th ed.). Wiley.
3. Kimball, R., & Ross, M. (2013). The data warehouse toolkit: The definitive guide to dimensional modeling (3rd ed.). Wiley.
4. Ponniah, P. (2011). Data warehousing fundamentals for IT professionals (2nd ed.). Wiley.
5. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data mining: Practical machine learning tools and techniques (4th ed.). Elsevier.

SURESH GYAN VIHAR UNIVERSITY
Accredited by NAAC with 'A+' Grade

SYLLABUS

(SEMESTER-V)

Program: Bachelor of Computer Applications

Course Name: Industrial Training / Internship	Course Code: D351
Semester: 5	Core/ Elective: Core
Teaching Scheme in Hrs (L:T:P): 0:0:0	Credits: 20
Type of course: Project	Total Contact Hours: 3
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Domain Specific Knowledge.

Course Objectives:

To provide industry exposure to the student, 6 months of training in a reputed software industry is mandatory for every student in the 4th Semester.

Course Content:

Topic and Contents
Training as per the industrial requirements.

Course outcomes:

Successful completion of this course makes students more employable, skilled and equipped with more knowledge.

SYLLABUS

(SEMESTER-VI)

Program: Bachelor of Computer Applications

Course Name: Introduction to Sales force	Course Code: DCA302
Semester: 6	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 4
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of computer applications and cloud concepts.

Course Objectives:

1. Understand the fundamentals of Sales force and its role in CRM.
2. Learn Sales force's data modeling and customization features.
3. Gain hands-on experience in managing and automating business processes using Sales force.
4. Explore the declarative and programmatic tools provided by Sales force.
5. Develop skills to build and manage reports, dashboards, and workflows.

Course Content:

Topic and Contents
BLOCK-1: Introduction to Salesforce
Unit 1: Overview of CRM and Sales force Unit 2: Sales force Architecture and Cloud Offerings Unit 3: Sales force Editions and Licensing
BLOCK-2: Salesforce Administration
Unit 4: User Management and Security Unit 5: Data Management (Objects, Fields, Relationships) Unit 6: Profiles, Roles, and Permissions
BLOCK-3: Salesforce Automation
Unit 7: Workflow Rules and Approval Processes Unit 8: Process Builder and Flow Automation Unit 9: Email Templates and Notifications

BLOCK-4: Reporting And Dashboards
Unit 10: Creating Reports (Standard and Custom) Unit 11: Building Dashboards and Data Visualization Unit 12: Report Types and Filters
BLOCK-5: Introduction to Salesforce Development
Unit 13: Overview of Apex and Visual force Unit 14: Introduction to Lightning Web Components (LWC) Unit 15: Basics of Sales force Integration (APIs and Connectors)

Course Outcomes:

1. Gain an understanding of Sales force CRM and its functionalities.
2. Manage users, security settings, and data effectively in Sales force.
3. Automate workflows and approval processes to enhance productivity.
4. Create insightful reports and dashboards to support decision-making.
5. Get an introduction to Sales force development and integration techniques.

Reference:

1. Salesforce. (2023). Salesforce: The customer success platform. Salesforce. https://www.salesforce.com
2. Jobber, D., & Lancaster, G. (2019). Selling and sales management (11th ed.). Pearson Education.
3. Zoltners, A. A., Sinha, P., & Lorimer, S. E. (2009). Building a winning sales force: Powerful strategies for driving high performance. AMACOM.
4. Buttle, F., & Maklan, S. (2019). Customer relationship management: Concepts and technologies (4th ed.). Routledge.
5. Kotler, P., & Keller, K. L. (2016). Marketing management (15th ed.). Pearson Education.

Program: Bachelor of Computer Applications

Course Name: Research and Intellectual Property Rights	Course Code: DCA304
Semester: 6	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 4
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic knowledge of academic research processes. Awareness of legal and ethical considerations in professional fields.

Course Objectives:

1. Understand the fundamentals of research methodology and its application.
2. Learn the essentials of intellectual property rights (IPR) and their importance in innovation.
3. Study the ethical and legal aspects of research and intellectual property.
4. Explore patent filing processes, copyrights, and trademarks.
5. Understand the commercialization of intellectual property and its impact on industries.

Course Content:

Topic and Contents
BLOCK-1: Introduction to Research Methodology
Unit 1: Basics of Research and Types of Research Unit 2: Research Design and Data Collection Methods Unit 3: Data Analysis and Report Writing
BLOCK-2: Introduction to Intellectual Property Rights (IPR)
Unit 4: Overview of IPR: Patents, Copyrights, and Trademarks Unit 5: Importance of IPR in Innovation and Creativity Unit 6: International IPR Systems and Agreement
BLOCK-3: Patents and Trademarks
Unit 7: Patent Search, Filing, and Grant Process Unit 8: Types of Trademarks and Their Registration Unit 9: Case Studies on Patents and Trademarks

BLOCK-4: Copyrights and Trade Secrets
Unit 10: Fundamentals of Copyright Laws Unit 11: Licensing and Infringement Issues Unit 12: Protection of Trade Secrets and Confidential Information
BLOCK-5: Ethics and Commercialization
Unit 13: Ethical Issues in Research and IPR Unit 14: Technology Transfer and Commercialization of IPR Unit 15: Case Studies on Ethical Dilemmas and Successful Commercialization

Course Outcomes:

1. Gain an understanding of research methodologies and their practical application.
2. Learn about the significance and types of intellectual property rights.
3. Acquire knowledge of the processes involved in securing and protecting intellectual property.
4. Understand the ethical implications and legal frameworks surrounding IPR.
5. Explore ways to commercialize research outputs and intellectual property.

Reference:

1. WIPO. (2020). What is intellectual property? World Intellectual Property Organization. https://www.wipo.int/about-ip/en/
2. Creswell, J. W., & Creswell, J. D. (2018). Research design: Qualitative, quantitative, and mixed methods approaches (5th ed.). SAGE Publications.
3. Bainbridge, D. (2019). Intellectual property (12th ed.). Pearson Education.
4. Ganguli, P. (2001). Intellectual property rights: Unleashing the knowledge economy. Tata McGraw-Hill Education.
5. Kothari, C. R., & Garg, G. (2019). Research methodology: Methods and techniques (4th ed.). New Age International Publishers.

Program: Bachelor of Computer Applications

Course Name: Data Science Basic	Course Code: DCA306
Semester: 6	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 4
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of mathematics, statistics, and programming concepts.

Course Objectives:

1. Understand the fundamental concepts and lifecycle of data science.
2. Learn to collect, clean, and preprocess data for analysis.
3. Study core techniques such as statistical modeling, machine learning, and data visualization.
4. Explore tools and frameworks for implementing data science projects.
5. Gain insights into ethical practices and applications of data science across industries.

Course Content:

Topic and Contents
BLOCK-1: Introduction to Data Science
Unit 1: Overview of Data Science and Its Applications Unit 2: Data Science Workflow and Lifecycle Unit 3: Tools and Technologies in Data Science (Python, R, etc.)
BLOCK-2: Data Collection and Preprocessing
Unit 4: Data Types and Sources Unit 5: Data Cleaning Techniques Unit 6: Data Transformation and Feature Engineering
BLOCK-3: Statistics and Data Analysis
Unit 7: Descriptive Statistics and Probability Theory Unit 8: Inferential Statistics and Hypothesis Testing Unit 9: Exploratory Data Analysis (EDA)

BLOCK-4: Machine Learning
Unit 10: Supervised Learning (Regression and Classification)
Unit 11: Unsupervised Learning (Clustering and Dimensionality Reduction)
Unit 12: Model Evaluation and Validation Techniques
BLOCK-5: Data Visualization and Ethics
Unit 13: Data Visualization Techniques (Using Tools Like Matplotlib, Tableau)
Unit 14: Interpreting and Presenting Results
Unit 15: Ethical Issues and Challenges in Data Science

Course Outcomes:

1. Develop a strong foundation in data science concepts and practices.
2. Gain proficiency in data preprocessing, statistical analysis, and visualization.
3. Apply machine learning techniques to solve real-world problems.
4. Learn to use modern tools and frameworks for data analysis and modeling.
5. Understand the ethical implications and responsibilities of a data scientist.

Reference:

1. Provost, F., & Fawcett, T. (2013). Data science for business: What you need to know about data mining and data-analytic thinking. O'Reilly Media.
2. Grus, J. (2019). Data science from scratch: First principles with Python (2nd ed.). O'Reilly Media.
3. McKinney, W. (2022). Python for data analysis: Data wrangling with pandas, NumPy, and Jupyter (3rd ed.). O'Reilly Media.
4. VanderPlas, J. (2016). Python data science handbook: Essential tools for working with data. O'Reilly Media.
5. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer.

Program: Bachelor of Computer Applications

Course Name: Data Science Basic Lab	Course Code: DCA352
Semester: 6	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 1
Type of course: LAB	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic knowledge of programming in Python or R.

Course Objectives:

1. Provide hands-on experience with data preprocessing and visualization.
2. Develop skills in implementing statistical analysis and machine learning algorithms.
3. Familiarize students with real-world datasets and practical problem-solving.
4. Enhance the ability to interpret results and build data-driven solutions.
5. Improve proficiency in using data science tools and frameworks.

Course Content:

Topic and Contents
Lab 1: Perform data cleaning, handling missing values, and outlier detection using Python libraries like Pandas and NumPy.
Lab 2: Implement data visualization techniques using Matplotlib and Seaborn (e.g., line graphs, bar charts, heatmaps).
Lab 3: Conduct exploratory data analysis (EDA) on a real-world dataset and summarize insights.
Lab 4: Apply statistical measures like mean, median, mode, variance, and correlation on sample datasets.
Lab 5: Develop a linear regression model and evaluate it using metrics like R-squared and Mean Squared Error (MSE).
Lab 6: Implement classification algorithms such as Logistic Regression and Decision Trees
Lab 7: Use clustering algorithms like K-Means and Hierarchical Clustering for unsupervised learning.
Lab 8: Solve dimensionality reduction problems using Principal Component Analysis (PCA).
Lab 9: Perform sentiment analysis on text data using Natural Language Processing (NLP) techniques.
Lab 10: Create and evaluate a complete machine-learning pipeline using Scikit-learn.

Course Outcomes:

1. Apply data preprocessing and analysis techniques to diverse datasets.
2. Implement statistical methods and machine learning models for practical problems.
3. Build, test, and evaluate supervised and unsupervised learning models.
4. Gain hands-on experience with data visualization tools and frameworks.
5. Develop end-to-end data science solutions to real-world problems.

Reference:

1. Provost, F., & Fawcett, T. (2013). Data science for business: What you need to know about data mining and data-analytic thinking. O'Reilly Media.
2. Grus, J. (2019). Data science from scratch: First principles with Python (2nd ed.). O'Reilly Media.
3. McKinney, W. (2022). Python for data analysis: Data wrangling with pandas, NumPy, and Jupyter (3rd ed.). O'Reilly Media.
4. VanderPlas, J. (2016). Python data science handbook: Essential tools for working with data. O'Reilly Media.
5. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer.

Program: Bachelor of Computer Applications

Course Name: Software Testing	Course Code: DCA308
Semester: 6	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 4
Type of course: Lecture+ Assignments	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic understanding of software development life cycle (SDLC) and programming.

Course Objectives:

1. Understand the principles and methodologies of software testing.
2. Learn to design effective test cases and strategies.
3. Familiarize with various testing techniques and tools.
4. Develop skills to identify, document, and resolve software defects.
5. Explore quality assurance processes and industry best practices.

Course Content:

Topic and Contents
BLOCK-1: Introduction to Software Testing
Unit 1: Basics of Software Testing Unit 2: Software Development Life Cycle (SDLC) and Testing Life Cycle (STLC) Unit 3: Types of Testing
BLOCK-2: Test Case Design and Management
Unit 4: Test Case Design Techniques Unit 5: Test Planning and Strategy Unit 6: Test Management and Execution
BLOCK-3: Functional and Non-Functional Testing
Unit 7: Black Box and White Box Testing Unit 8: Performance Testing Unit 9: Security and Usability Testing

BLOCK-4: Automated Testing
Unit 10: Introduction to Test Automation Unit 11: Tools for Test Automation (e.g., Selenium, JUnit) Unit 12: Writing and Executing Automated Test Scripts
BLOCK-5: Quality Assurance and Reporting
Unit 13: Defect Lifecycle and Reporting Unit 14: Software Quality Standards (ISO, CMMI) Unit 15: Best Practices in Software Quality Assurance

Course Outcomes:

1. Gain a comprehensive understanding of software testing concepts and methodologies.
2. Design, execute, and manage effective test plans and cases.
3. Apply various testing techniques to ensure software reliability and performance.
4. Utilize automated testing tools to streamline the testing process.
5. Enhance software quality through systematic defect detection and resolution.

Reference:

1. Myers, G. J., Sandler, C., & Badgett, T. (2011). The art of software testing (3rd ed.). Wiley.
2. Kaner, C., Falk, J., & Nguyen, H. Q. (1999). Testing computer software (2nd ed.). Wiley.
3. Pressman, R. S. (2014). Software engineering: A practitioner's approach (8th ed.). McGraw-Hill Education.
4. Ammann, P., & Offutt, J. (2016). Introduction to software testing (2nd ed.). Cambridge University Press.
5. Copeland, L. (2003). A practitioner's guide to software test design. Artech House.

Program: Bachelor of Computer Applications

Course Name: Software Testing Lab	Course Code: DCA354
Semester: 6	Core / Elective: Core
Teaching Scheme in Hrs (L:T:P): 3:0:0	Credits: 1
Type of course: LAB	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic programming skills and understanding of software development processes.

Course Objectives:

1. Provide hands-on experience in designing and executing test cases.
2. Familiarize with automated testing tools and frameworks.
3. Enable understanding of defect identification, documentation, and resolution.
4. Explore functional and non-functional testing techniques.
5. Build the ability to apply testing methodologies in real-world scenarios.

Course Content:

Topic and Contents
Lab 1: Write and execute manual test cases for a simple application (e.g., login functionality).
Lab 2: Perform boundary value analysis and equivalence partitioning for input validation.
Lab 3: Conduct black-box and white-box testing on a sample program.
Lab 4: Design test scenarios for integration testing of a multi-module application.
Lab 5: Execute system testing on a real-world application and generate reports.
Lab 6: Automate functional testing using Selenium IDE and write simple test scripts.
Lab 7: Perform performance testing using tools like JMeter (e.g., load testing for a web application)
Lab 8: Conduct security testing on a sample web application to identify vulnerabilities.
Lab 9: Write automated test cases using a unit testing framework (e.g., JUnit or TestNG).
Lab 10: Document defects with severity and priority using a defect tracking tool (e.g., Bugzilla, Jira).

Course Outcomes:

1. Gain practical experience in creating and executing various test cases.
2. Apply automated testing tools for functional and performance testing.
3. Identify and document defects effectively using professional tools.
4. Perform comprehensive testing to ensure software quality.
5. Develop industry-relevant skills in software testing and quality assurance.

Reference:

1. Myers, G. J., Sandler, C., & Badgett, T. (2011). The art of software testing (3rd ed.). Wiley.
2. Kaner, C., Falk, J., & Nguyen, H. Q. (1999). Testing computer software (2nd ed.). Wiley.
3. Pressman, R. S. (2014). Software engineering: A practitioner's approach (8th ed.). McGraw-Hill Education.
4. Ammann, P., & Offutt, J. (2016). Introduction to software testing (2nd ed.). Cambridge University Press.
5. Copeland, L. (2003). A practitioner's guide to software test design. Artech House.

Program: Bachelor of Computer Applications

Course Name: Projects	Course Code: DCA356
Semester: 6	Core/Elective: Core
Teaching Scheme in Hrs(L: T:P): 0:0:2	Credits: 1
Type of course: Projects	Total Contact Hours: 10
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-Requisites:

Basic knowledge of programming language and understanding of fundamental programming concepts.

Course Objective:

1. Application of Basic Syntax and Concepts
2. Implementation of Object-Oriented Programming
3. Problem-Solving and Algorithm Development
4. Data Management and File Handling
5. User Interface Design

Course Content:

Topic and Contents
Lab 1: To do List Application
Lab 2: Unit Converter
Lab 3: Number Guessing Game
Lab 4: Basic Banking System
Lab 5: Expense Tracker
Lab 6: Library Catalog System
Lab 7: Flashcard Quiz App
Lab 8: Flashcard Quiz App
Lab 9: Random Quote Generator
Lab 10: Simple Chatbot

Course Outcomes:

1. Demonstrated Proficiency in Programming Concepts
2. Development of Problem-Solving Skills
3. Experience with Software Development Practices
4. User Interface Design and Usability Awareness
5. CollaborativeProjectWorkSuccessfulImplementationofUser-CentricConsoleApplications.

Program: Master of Computer Applications

Course Name: Seminar	Course Code: DCA358
Semester: 6	Core / Elective:
Teaching Scheme in Hrs (L: T:P): 0:0:2	Credits: 1
Type of course: Lab	Total Contact Hours: 12
Continuous Internal Evaluation: 30 Marks	ESE: 70 Marks

Pre-requisites:

Basic knowledge of programming, databases, and web technologies.

Course Objectives:

1. To introduce students to real-world software development methodologies.
2. To develop skills in full-stack and cross-platform application development.
3. To apply data analysis and machine learning techniques on practical datasets.
4. To design and deploy secure and scalable web, mobile, and IoT applications.
5. To provide hands-on experience in emerging technologies like blockchain and cloud computing.

Course Content:

Topic and Contents
Lab 1: Build a full-stack web application using MERN (MongoDB, Express.js, React, Node.js).
Lab 2: Develop across-platform mobile app using Flutter or React Native with backend integration.
Lab 3: Perform data analysis using Python and libraries like Pandas and Matplotlibon a real dataset.
Lab 4: Train a machine earning model with sci-kit-learn.
Lab 5: Create a cyber security awareness web-tool.
Lab 6: Deploy an app on AWS or Azure.
Lab 7: Develop a simple Ethereum block chain application.
Lab 8: Create and document a RESTful API.
Lab 9: Build an e-commerce website with payment integration.
Lab 10: Create an IoT application with sensor data visualization.

Course outcomes:

1. Students will be able to build and deploy full-stack and cross-platform applications.
2. Students will perform effective data analysis and visualize results using Python tools.
3. Students will train and evaluate machine learning models using modern libraries.
4. Students will demonstrate the ability to create secure web tools and integrate payment systems.
5. Students will implement and document RESTful APIs and deploy projects on cloud platforms.

Procedure for Admission, Curriculum Transaction and Evaluation

The proposed program in ODL mode will be conducted by CDOE-SGVU with the support of various departments of the University. Eligibility criteria, course structure, detailed curriculum, duration of program and evaluation criteria shall be approved by Board of Studies and Academic Council, SGVU, Jaipur which are based on UGC guidelines for the program which comes under the purview of ODL and mode for award of Degree. Details of Procedure for admission in which eligibility criteria for admission and fee structure of the course, Curriculum includes Program delivery, norms for delivery of courses in ODL mode, use of IT services to academic support services, course design academic calendar and Evaluation which includes Distribution of Marks in Continuous Internal Assessments, Minimum Passing criteria and system of Grading formats are given in detail as under.

Procedure for Admission

Students who will seek admission in BCA (Bachelor of Computer Applications) program to apply through its website www.sgvu.edu.in.

Minimum Eligibility Criteria for Admission

The minimum eligibility criteria for admission in ODL BCA (Bachelor of Computer Applications) program is a passing 12th Class from any Recognized Board with Mathematics as main subjects.

Program Fee and Financial Assistance Policy

Program fees for students for proposed BCA in various streams offered by CDOE-SGVU Jaipur is Rs. 21000 per year, where Rs.18000 is the tuition fees and Rs. 3000 is examination fees.

Program Delivery

The curriculum will be delivered through the Self Learning Materials (SLMs) supported by various learning resources including audio-video aids.

Academic Calendar

S. No.	Name of the Activity	Tentative months schedule during year			
		From (Month)	To (Month)	From (Month)	To (Month)
1	Admission	Jul	Sep	Jan	Feb
2	Assignment Submission (if any)	Oct	Nov	April	May
3	Evaluation of Assignment	Nov	Dec	May	June
4	Examination	Dec	Jan	June	Jul
5	Declaration of Result	Feb	Mar	Aug	Sep
6	Re–registration	Jan	Feb	Jul	Sep
7	Distribution of SLM	Jul	Sep	Jan	Feb
8	Contact Program, Counselling, Practical, etc.)	Nov	Dec	May	June

Evaluation

The evaluation shall include two types of assessments-

1. Continuous Assessment in the form of assignments (30% Weightage).
2. End Semester Examination, which will be held at the SGVU campus (70% Weightage).

Minimum Passing percentage

The marks of both the components (continuous assessment & end semester exam evaluation) of a course shall be added to get total marks out of 100. Minimum passing marks in end semester examinations/overall in each course shall be as follows:

- For UG program:

End semester examination: 30%, Total Marks: 37%

- For PG program:

End semester examination: 40%, Total Marks: 46%

Marks and Grades

Grades & Grade Points

a. At the end of the Semester / Year every student is assigned a 'Letter Grade' based on his/her performance over the semester in all courses for which he/she had registered.

b. The letter grade and grade point indicate the results of quantitative and qualitative assessment of the student's performance in a course.

c. There are seven letter grades: S, A+, A, B+, B, C, D, F that have grade points with values distributed on a 10-point scale.

Requirement of the Laboratory Support and Library Resources

The university is having the state-of-the-art computer labs and the associated peripherals to support any number of students at a given point of time, the students can learn and carry out in length academic & research activities at the computer labs in the SGVU.

Library Resources

CDOE-SGVU has excellent library with all the books required for the course learning and reference books for the course of BCA (Bachelor of Computer Applications). Adequate online learning links and e-learning materials will also be provided to students which will be support for students in their learning cycle.

Cost Estimate of the Program and the Provisions

The Estimate of approximate Cost & Budget could be as follows (all figures on Annual basis):

1. Salaries: Rs. 60,00,000/- (Approx).
2. Travel: Rs. 30,000/- (Approx).
3. Seminars: Rs. 2,00,000/- (Approx).
4. SLM Preparation, Printing, Distribution: Rs. 3,00,000/- (Approx).
5. Library & e-resources (including membership like DELNET): 3,50,000/- (Approx).
6. Courier/Transportation: Rs. 50,000/- (Approx).
7. Infrastructure: Rs. 5,00,000/- (Approx)
8. Computer Labs & Leased Line: Rs. 1,00,000/- (Approx)
9. E-contents development: 8,00,000 (Approx)
10. LMS & its Maintenance (including server): 2,00,000 (Approx.)

Quality Assurance Mechanism and Expected Program Outcomes

- The quality of the program depends on the course curriculum and syllabus which meets the requirement of the industry and creates the skillful learning in the students. The ultimate aim of BCA (Bachelor of Computer Applications) program in ODL Mode is to enhance skill soft he learners as managers, entrepreneurs and seeing them excel in their profession and meeting global standards too by upgrading their career opportunities.
- The CDOE, SGVU, Jaipur has constituted Centre for Internal Quality Assurance (CIQA). The CIQA will do periodic assessment of the online learning course material and audio video tutorials to ensure quality of learning and time to time changes are made as per the course requirement.
- The CIQA will also access the quality of assignments, quizzes and end term assessment time to time and required changes will be assured. CIQA will assure that the learning is made a truly global experience for the learner along with inculcation of required skills in the learner as expected program outcome with CDOE, SGVU, Jaipur.
- The university will work continuously for the betterment of processes, assessments, teaching methodology, e-learning material improvisation as per four quadrant approach and implementation of the same as per New Education Policy. The University is committed to deliver the best education in all the learning modes with adherence to NEP, UGC and other regulatory guidelines in truly global sense.